



How-to Guide

# How To Implement Object Links for SAP Portfolio and Project Management

Version 1.1 – December 2010



© Copyright 2010 SAP AG. All rights reserved

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, Clear Enterprise, SAP BusinessObjects Explorer, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP France in the United States and in other countries.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

#### **Disclaimer**

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

#### **Documentation in the SAP Service Marketplace**






You can find this documentation at the following address:

<http://service.sap.com/instguides>

## Typographic Conventions

Type Style	Represents
<i>Example Text</i>	Words or characters quoted from the screen. These include names, screen titles, pushbutton labels, menu names, menu paths, and menu options. Cross-references to other documentation.
<b>Example text</b>	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	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 text	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.
<b>Example text</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<b>&lt;Example text&gt;</b>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXTKEYS	Keys on the keyboard, for example, F2 or ENTER.

## Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

# Table of Contents

<b>1</b>	<b>Instructions .....</b>	<b>6</b>
<b>2</b>	<b>Using This Document.....</b>	<b>9</b>
<b>3</b>	<b>Functions .....</b>	<b>10</b>
3.1	General .....	10
3.1.1	Key and Identification .....	10
3.1.2	Description.....	11
3.1.3	Search Help Fields.....	12
3.1.4	Proxy Classes .....	13
3.1.5	Table/View in SAP System per Object Type and Method.....	13
3.1.5.1	Existence Check.....	14
3.1.5.2	Consistency Check .....	14
3.1.5.3	Search Help.....	14
3.1.5.4	Data Retrieval (Evaluation/Detail Screen) .....	14
3.1.5.5	Date Retrieval .....	14
3.1.5.6	Language-Dependent Description .....	14
3.1.5.7	Dependencies .....	14
3.1.5.8	Occurrence in the Sample Object Types.....	15
3.1.6	Search Help.....	16
3.2	Existence Check .....	17
3.2.1	Enhancement of the Existence Check.....	17
3.2.2	Method IF_DPR_OBL_PROXY~CHECK_EXISTENCE .....	18
3.2.3	Method IF_OLR3_PROXY~CHECK_EXISTENCE.....	19
	Consistency Check .....	20
3.2.4	Enhancement of the Consistency Check .....	20
3.2.5	Method IF_DPR_OBL_PROXY~CHECK_CONSISTENCY.....	21
3.2.6	Method IF_OLR3_PROXY~CHECK_CONSISTENCY .....	22
3.3	Search Help .....	23
3.3.1	Enhancement of the Search Help.....	23
3.3.2	Method IF_DPR_OBL_PROXY~SEARCH_OBJECTS .....	25
3.3.3	Method IF_OLR3_PROXY~SEARCH_OBJECTS.....	26
3.4	Data Retrieval.....	26
3.4.1	Data Retrieval Enhancement .....	27
3.4.2	Method IF_DPR_OBL_PROXY~GET_DATA.....	28
3.4.3	Method IF_OLR3_PROXY~GET_DATA .....	29
3.5	Date Retrieval.....	30
3.5.1	Date Retrieval Enhancement .....	30
3.5.2	Method IF_OLR3_PROXY~GET_DATES .....	31

3.6	Calling an SAP Transaction.....	32
3.6.1	Service for an SAP Transaction .....	32
3.6.2	Service Parameters .....	33
	3.7.2.1 Name of a Parameter in a Start URL .....	33
	3.7.2.2 Value of a Parameter in a Start URL .....	33
4	Attributes for Evaluations.....	35
5	User-Defined Object Links.....	36
5.1	Using the Object Types Delivered As Examples.....	36
5.2	Copying an Object Type from the Sample Development .	36
5.3	Using an Object Type from the Sample Development in a Modified Form .....	37
5.3.1	Key, Identification, and Description.....	37
5.3.2	Search Fields .....	38
5.3.3	Proxy Classes .....	38
5.3.4	Table/View in SAP System per Object Type and Method.....	39
5.4	Copying an Object Type from the Sample Development and Adapting It to a Semantically Similar SAP Object Type .....	39
5.5	Creating Without a Template .....	39

# 1 Instructions

SAP delivers a sample implementation with SAP Portfolio and Project Management for linking objects with the following SAP object types:

Example of SAP object types for Project Management:

SAP Object Type	Object Type in SAP Portfolio and Project Management
Project definition	0PROJECTDEF
SAP DMS document	0DOCUMENT

You can find this sample implementation in Customizing for *SAP Portfolio and Project Management* under Project Management → Structure → Define Object Types for Object Links.

Example of SAP object types for Portfolio Management:

SAP Object Type	Object Type in SAP Portfolio and Project Management
PS Project Definition	0RPMPSPROJECT
DMS Document	0RPMDOCUMENT

You can find this sample implementation in Customizing for *SAP Portfolio and Project Management* under Portfolio Management → Base System Interfaces → Application Object Settings → Define Object Link Types.

To use these object links with their preconfigured settings, you have to carry out the following steps:

1. We recommend that you install the software component SAP ERP Central Component 6.0 (SAP ECC 6.0) in the required SAP system at least. Note the following:
  - The installation of a corresponding plug-in (at least PI 2002.2) is required for previous SAP R/3 releases.
  - A higher SAP ERP release is required for the usage of certain newer object link types:
    - The object link type 0CONTROLPLAN (Control Plan (QM)) requires SAP ERP 6.0 enhancement package (EHP) 3.
    - The following object link types require SAP ERP 6.0 EHP4:

Object Link Type	Description
0IPPECMPNODE	(iPPE - CMP Node (Product Structure))
0IPPECMPVARIANT	(iPPE - CMP Variant (Product Structure))
0IPPECONVARIANT	(iPPE - CON Variant (Concept))
0IPPEFOCUS	(iPPE - Focus Master Record)

2. Define the logical connection for this SAP system in SAP Portfolio and Project Management using transaction SM59.



You should be able to log on to the SAP system remotely from SAP Portfolio and Project Management without using a logon screen. You can do this if one of the following is true:

- The systems have a trusted system relationship with one another.
- The logon is set up in transaction SM59 using Single Sign-On (SSO).
- A fixed user with password has been entered in transaction SM59.

3. **Up to SAP NetWeaver Release 2004:**

Install an SAP Internet Transaction Server (ITS) (at least 6.20) for the required SAP system.



The system only supports calling SAP transactions for SAP object links via an ITS. For more information about installing an ITS, see the documentation on SAP Service Marketplace at [service.sap.com/instguides](http://service.sap.com/instguides) → *SAP NetWeaver* → *SAP NetWeaver 2004*.

**As of SAP NetWeaver Release 700:**

Activate the internal SAP Internet Transaction Server (ITS).



The system only supports calling SAP transactions for SAP object links via an internal ITS. For more information about installing an internal ITS, see the documentation on SAP Service Marketplace at [service.sap.com/instguides](http://service.sap.com/instguides) → *SAP NetWeaver* → *SAP NetWeaver 7.0 (2004s)*.

4. For all required object types, select the *Active* indicator and enter the logical destination (see step 2) and the Web server address of the ITS (see step 3). For Project Management, you make these settings in the Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Structure* → *Define Object Types for Object Links*.

For Portfolio Management, you make these settings in the Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*

5. For Project Management object types, make the following settings in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*:
  - In the *Standard Consistency Checks per Object Type* view, check which consistency checks are active or inactive in standard Customizing for each object type that is used.
    - Deselect the *Active* indicator for all standard checks you do not require.
    - Select the *Active* indicator for all standard checks you require.
  - In the *Frames per Object Type* view, deselect the *Active* indicator for all frames you do **not** want to display in the SAP Portfolio and Project Management detail screen. Do this for each object type that is used.

- In the *Fields per Frame* view, deselect the *Active* indicator for all fields you do **not** want to display in a frame in the SAP Portfolio and Project Management detail screen. Do this for each object type that is used.

For Portfolio Management object types, these settings can be made under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → → *Define Object Link Types*.

6. Select one of the supported date types in the user settings for the Project Management application.

## 2 Using This Document

- If you want to use only one instance per SAP object type delivered, only section 1 is relevant for you.
- If you want to use more than one instance of a delivered SAP object type, section 1 and section 5.2 are relevant. This is the case, for example, if you use different SAP destinations or if you use different Web servers, or if you use both different SAP destinations and Web servers for each concrete object type.
- If you want to use an SAP object type in its modified form by changing a technical setting in the copied object type, see the different parts of section 3.1 and see section 5.3 (depending on the technical setting that was changed).
- If you want to use an SAP object type in a semantically similar way, the following sections are relevant:
  - Section 3
  - Section 4
  - Section 5.4
  - Section 6
- If you want to develop your own object types without using a template of an SAP object type, a framework, or the infrastructure of this sample implementation, this document is not relevant for you (compare with section 5.5). In this case, note the general documentation for object links in SAP Portfolio and Project Management.

## 3 Functions

### 3.1 General

You can make the following technical settings for a Project Management object link type in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to Project Management* → *External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*:

- Key and identification (see section 3.1.1)
- Description (see section 3.1.2)
- Search help fields (see section 3.1.3)
- Proxy class in SAP Portfolio and Project Management and in the SAP system (see section 3.1.4 and section 3.2 and the pages that follow for the methods of the proxy classes).
- Table or view in the SAP system per object type and method (see section 3.1.5)
- Destination (see section 1)
- Web server (see section 1)

For Portfolio Management object link types, these settings can be made under the following customizing activity:

*Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*.



The system reads the general SAP Portfolio and Project Management settings for the object type with the method GET\_OBJECT\_CAPABILITIES of the class CL\_DPR\_OBL\_REPOSITORY. The system reads the special settings used for this sample implementation for the object type with the methods of the class CL\_DPR\_OLR3\_REPOSITORY.

#### 3.1.1 Key and Identification

An SAP object can be identified either exclusively by external key fields or by external and internal key fields. The user only sees the external key fields on the user interface.

If an SAP object is only identified by external key fields, only one key is defined for the object type. The key contains the name of the DDIC structure in which the external key fields are defined.

If an SAP object is identified by external and internal key fields, a single key and a single identification are defined for the object type. The key contains the name of the DDIC structure in which the internal key fields are defined. The identification contains the name of the DDIC structure in which the external key fields are defined.

You can create and display DDIC structures in transaction SE11.

Example of object type keys and identifications for Project Management:

Object Type	Key	Identification
0PROJECTDEF	OLR3_TS_PROJ_INT_ID	OLR3_TS_PROJ_EXT_ID
0DOCUMENT	OLR3_TS_DRAW_INT_ID	

For the object types in which only one key is defined, the system includes the structure OLR3\_TS\_XXX\_EXT\_ID each time in the DDIC structure OLR3\_TS\_XXX\_INT\_ID.

Each DDIC structure contains the following key fields and identification fields:

Object Type	Fields in the DDIC Structure in Key	Fields in the DDIC Structure in Identification
0PROJECTDEF	PSPNR Internal Number Definition OLR3_GUID if it exists: GUID	PSPID Name Project Definition
0DOCUMENT	DOKNR Document Number DOKAR Document Type DOKTL Document Part DOKVR Document Version	

The field names used here correspond to the database fields or view fields in the SAP system.

Example of Portfolio Management object type keys and identifications:

Object Type	Key	Identification
0RPMPSPROJECT	OLR3_TS_PROJ_INT_ID	OLR3_TS_PROJ_EXT_ID
0RPMDOCUMENT	OLR3_TS_DRAW_INT_ID	OLR3_TS_DRAW_EXT_ID

For Portfolio Management object links, it is mandatory to define both the key and identification structures.

### 3.1.2 Description

You cannot explicitly define the origin of the description from the SAP system for an object type in Customizing. Therefore, in this sample implementation, an implicit assignment to a corresponding DDIC structure for the object type is a prerequisite and we use this.

To determine the name of the DDIC structure, replace "INT\_ID" with "DESC" in the DDIC structure of the key.

Object Type	Name of DDIC Structure	Fields of the DDIC Structure of the Description
0PROJECTDEF	OLR3_TS_PROJ_DESC	POST1
0DOCUMENT	OLR3_TS_DRAW_DESC	DKTXT (language-dependent) LANGU
0RPMPSPROJECT	OLR3_TS_PROJ_DESC	POST1
0RPMDOCUMENT	OLR3_TS_DRAW_DESC	DKTXT (language-dependent) LANGU

Language-dependent descriptions require the field name of the language as the second field. The field names used here correspond to the database fields or view fields in the SAP system.

### 3.1.3 Search Help Fields

You can search for an SAP object in SAP Portfolio and Project Management. The corresponding help search fields are in the DDIC structure that is assigned to the object type in the *Search Fields* field in Customizing. For more information on Project Management object links, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* → *Object Types for Object Links*.

For more information about Portfolio Management object links, see Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types* → *Settings for Object Link Type*.

Object Type	Search Fields	Search Help Fields
0PROJECTDEF	OLR3_TS_PROJ_SHLP	PSPID Name Project Definition POSTU Description VERNR Number of Responsible VERNA Name of Responsible ASTNR Number of Requester ASTNA Name of Requester
0DOCUMENT	OLR3_TS_DRAW_SHLP	DOKNR Document Number DOKAR Document Type DOKTL Document Part DOKVR Document Version DKTXT_UC Language-Dependent Description
0RPMSPROJECT	OLR3_TS_PROJ_SHLP	PSPID Name Project Definition POSTU Description VERNR Number of Responsible VERNA Name of Responsible ASTNR Number of Requester ASTNA Name of Requester
0RPMDOCUMENT	OLR3_TS_PROJ_SHLP	DOKNR Document Number DOKAR Document Type DOKTL Document Part DOKVR Document Version DKTXT_UC Language-Dependent Description

The field names used here correspond to the database fields or view fields in the SAP system. If the description is language-dependent and is part of the search criterion, the search in the SAP system takes place with the SAP Portfolio and Project Management logon language.

### 3.1.4 Proxy Classes

The program logic is encapsulated in SAP Portfolio and Project Management and in the SAP system in the corresponding proxy class of the current object type:

Object Type	Name of Proxy Class in SAP Portfolio and Project Management
0PROJECTDEF	CL_DPR_OLR3_PROJECTDEFINITION
0DOCUMENT	CL_DPR_OLR3_DOCUMENT
0RPMPSPROJECT	CL_DPR_OLR3_PROJECTDEFINITION
0RPMDOCUMENT	CL_DPR_OLR3_DOCUMENT



Each proxy class in SAP Portfolio and Project Management implements the interface IF\_DPR\_OBL\_PROXY.

Object Type	Name of Proxy Class in the SAP System
0PROJECTDEF	CL_R3_OLR3_PROJECTDEFINITION
0DOCUMENT	CL_R3_OLR3_DOCUMENT
0RPMPSPROJECT	CL_R3_OLR3_PROJECTDEFINITION
0RPMDOCUMENT	CL_R3_OLR3_DOCUMENT



Each proxy class in the SAP system implements the interface IF\_R3\_OLR3\_PROXY.

### 3.1.5 Table/View in SAP System per Object Type and Method

The following methods are available in the SAP system for the data origin of each SAP object type represented by an object type:

- Existence check
- Consistency check
- Search help
- Data retrieval
- Date retrieval

- Language-dependent description

The source of the data is always a database table or a database view in the SAP system, irrespective of the method.

### 3.1.5.1 Existence Check

The existence check in the SAP system takes place in the table or view you enter here with the fields described in section 3.1.1. It only makes sense to use one strategy for this method.

### 3.1.5.2 Consistency Check

The consistency check in the SAP system takes place with the object you selected in the SAP system from the table or view entered here. It only makes sense to use one strategy for this method.

### 3.1.5.3 Search Help

Depending on the concrete characteristics of the search help criteria, you have to use different search help views in the SAP system. Therefore, you have to use your own strategy for each view when using this method. When you carry out the search, you have to select a single strategy that matches the concrete characteristics of the search help criteria.

### 3.1.5.4 Data Retrieval (Evaluation/Detail Screen)

Data retrieval occurs in an evaluation of the Project Management and when calling the detail screen of the linked object. If more than one source exists from which the system reads the data, you have to create your own strategy for each source. When the system retrieves data it reads all strategies and consequently uses all tables or views of the object type as the generic data source.

### 3.1.5.5 Date Retrieval

Date retrieval is required for the date graphic in Project Management. In both of the named examples (project definition and SAP DMS document), date retrieval is only characterized for the SAP object type *project definition* (see section 3.6). It only makes sense to use one strategy for this method.

### 3.1.5.6 Language-Dependent Description

This method is only relevant for SAP object types that have a language-dependent description. This is the case for the SAP object type *document* in the named examples (project definition and SAP DMS document). It only makes sense to use one strategy for this method.

### 3.1.5.7 Dependencies

It only makes sense to use the specified table or view in the SAP system if the key fields (see section 3.1.1) you specified in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Structure* → *Define Object Types for Object Links* exist in the table or view and are unique.

- If only one external key exists for an object type, it must always be unique.
- If an external and an internal key exist for an object type, the external key fields for the existence check and search help must be unique. For all other methods, the internal key fields must be unique.

The following table shows in which field in Customizing the unique key must exist.

Method	Object with External Key	Object with External and
--------	--------------------------	--------------------------

	Fields	Internal Key Fields
Existence check	Key	Identification and key
Consistency check	Key	Key
Search help	Key	Identification
Data retrieval	Key	Key
Date retrieval	Key	Key
Language-dependent description	Key	Key

### 3.1.5.8 Occurrence in the Sample Object Types

M = Method

S = Strategy

Object Type	M	S	Table or View in SAP ERP Central Component (SAP ECC)	Explanation
0PROJECTDEF	1	1	V_OLR3_PROJ	View of the PROJ table with the internal and external key fields, the description, and the search help fields
0PROJECTDEF	2	1	PROJ	PROJ table
0PROJECTDEF	3	1	V_OLR3_PROJ	View of the PROJ table with the internal and external key fields, the description, and the search help fields
0PROJECTDEF	4	1	PROJ	PROJ table
0PROJECTDEF	5	1	PROJ	PROJ table
=====	==	=	=====	=====
0DOCUMENT	1	1	V_OLR3_DRAW	View of the DRAW table with the key fields and the search help fields without a description
0DOCUMENT	2	1	DRAW	DRAW table
0DOCUMENT	3	1	V_OLR3_DRAW	View of the DRAW table with the key fields and the search help fields without a description
0DOCUMENT	3	2	DRAT	DRAT table for the search help fields with a description
0DOCUMENT	4	1	DRAW	DRAW table
0DOCUMENT	6	1	DRAT	DRAT table
=====	==	=	=====	=====
0RPMPSPROJECT	1	1	PROJ	PROJ table

ORPMPSPROJECT	2	1	PROJ	PROJ table
ORPMPSPROJECT	3	1	V_OLR3_PROJ	View of the PROJ table with the internal and external key fields, the description, and the search help fields
ORPMPSPROJECT	4	1	PROJ	PROJ table
ORPMPSPROJECT	5	1	PROJ	PROJ table
=====	==	=	=====	=====
ORPMDOCUMENT	1	1	V_OLR3_DRAW	View of the DRAW table with the key fields and the search help fields without a description
ORPMDOCUMENT	2	1	DRAW	DRAW table
ORPMDOCUMENT	3	1	V_OLR3_DRAW	View of the DRAW table with the key fields and the search help fields without a description
ORPMDOCUMENT	3	2	DRAT	DRAT table for the search help fields with a description
ORPMDOCUMENT	4	1	DRAW	DRAW table
ORPMDOCUMENT	6	1	DRAT	DRAT table

### 3.1.6 Search Help

A view to configure search helps is available for the following object types:

Object Type	Search Help Name	Search Structure
ORPMPLMCOREMAT	/PLMB/MAT_ELM_MAT1W_E /PLMB/MAT_PLANT_ESH_DMZ	OLR3_TS_MARC_DB_SHLP OLR3_TS_MARC_ES_SHLP
ORPMPLMCORERCP	/PLMB/RCP_ELM_DMZ_DB /PLMB/RCP_ELM_DMZ_ES	OLR3_TS_PLM_RCP_SHLP OLR3_TS_PLM_RCP_SHLP
0PLMCOREMAT	/PLMB/MAT_ELM_MAT1W_E /PLMB/MAT_PLANT_ESH_DMZ	OLR3_TS_MARC_DB_SHLP OLR3_TS_MARC_ES_SHLP
0PLMCORERCP	/PLMB/RCP_ELM_DMZ_DB /PLMB/RCP_ELM_DMZ_ES	OLR3_TS_PLM_RCP_SHLP OLR3_TS_PLM_RCP_SHLP

You can select any one of the search helps available for an object type to be active. Select the enterprise search help if it is available on the target system; else use the database search help.

This search help can only be configured for object types belonging to the groups PLM\_CORE\_MARA and PLM\_CORE\_RCP.

## 3.2 Existence Check

When you create an object link in SAP Portfolio and Project Management with an SAP object type, the system only performs the existence check in the SAP system if you selected the *Existence Check* indicator in the object type. In this case, SAP Portfolio and Project Management calls the CHECK\_EXISTENCE method of the relevant SAP Portfolio and Project Management proxy class.

### 3.2.1 Enhancement of the Existence Check

#### Purpose



This section is aimed at SAP Portfolio and Project Management users with a basic knowledge of the programming environment in the SAP system.

You can enhance the existence check individually for the following interfaces using the CHECK\_EXISTENCE method:

- IF\_DPR\_OBL\_PROXY in the DEVELOPMENT\_PROJECTS\_OBJ\_LINK package
- IF\_OLR3\_PROXY in the R3\_OBJECTLINK\_R3 package

See the *Prerequisites* for the Customizing settings you need to make in the Customizing activity *Make Settings for the Linked SAP Objects* to be able to use these individual enhancements.

#### Prerequisites

- You have implemented the interface method IF\_OLR3\_PROXY~CHECK\_EXISTENCE in the proxy class of an SAP object link in the SAP system.
- You have implemented the interface method IF\_DPR\_OBL\_PROXY~CHECK\_EXISTENCE in the proxy class of an SAP object link in the SAP Portfolio and Project Management system.
- You have defined a DDIC structure for the object type of an SAP object link for the *Existence Check* method in the SAP Portfolio and Project Management system.

You have assigned the name of the corresponding proxy class in the SAP system to the object type of an SAP object link.

For more information about these settings for Project Management object links, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* in the description of the *Table/View in SAP System per Object Type and Meth.* view.

For Portfolio Management object links, see Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types* in the description of the view *Table/View in SAP System per Object Type and Meth.*

#### Procedure

We recommend that you note the following points when enhancing the existence check:

- Use the class builder in the SAP system to create the proxy class for a new SAP object link as a copy of an existing proxy class from the sample implementation in which the existence check has already been implemented. Then change the implemented interface method IF\_OLR3\_PROXY~CHECK\_EXISTENCE to meet your requirements.
- Do the same in SAP Portfolio and Project Management; copy an existing proxy class from the sample implementation in which the existence check has already been implemented. Then tailor the implemented interface method IF\_DPR\_OBL\_PROXY~CHECK\_EXISTENCE to meet your requirements.
- Create a new object type for SAP object links as a copy of an existing object type from the sample implementation. For object links from Project Management, copy the Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*, and copy all dependent entries. For object links from Portfolio Management, copy the Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*.

Afterwards you can change the settings in the lower-level maintenance views.

### 3.2.2 Method IF\_DPR\_OBL\_PROXY~CHECK\_EXISTENCE

#### Use

The interface method IF\_DPR\_OBL\_PROXY~CHECK\_EXISTENCE has been implemented in the SAP Portfolio and Project Management system in the following proxy classes of the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP Portfolio and Project Management System
Project definition	0PROJECTDEF	CL_DPR_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_DPR_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_DPR_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_DPR_OLR3_DOCUMENT

#### Interface of the Interface Method

Import Parameter	Description
IV_OBJECT_TYPE	Object type of linked object
Change Parameter	Description
CV_EXTERNAL_ID	String with the key fields of the identification (if it exists, see sections 3.1.1 and 3.1.5.7)
CV_OBJECT_KEY	String with the key fields of the key (see sections 3.1.1 and 3.1.5.7)
Export Parameter	Description
EV_DESCRIPTION	Description of linked object
EV_SYSTEM_ERROR	Type of system error when accessing the object

## Procedure

We recommend that you note the following points during implementation:

- In the SAP Portfolio and Project Management system, use the CHECK\_EXISTENCE method of the class CL\_DPR\_OLR3\_METHODS in the interface method IF\_DPR\_OBL\_PROXY~CHECK\_EXISTENCE to call the RFC function module OLR3\_RFC\_CHECK\_EXISTENCE in the SAP system.
- Implement a semantic here for the decision as to whether the description of the relevant SAP object type is language-dependent or not.

### 3.2.3 Method IF\_OLR3\_PROXY~CHECK\_EXISTENCE

## Use

The interface method IF\_OLR3\_PROXY~CHECK\_EXISTENCE has been implemented in the SAP system in the following proxy classes of the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP System
Project definition	0PROJECTDEF	CL_R3_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_R3_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_R3_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_R3_OLR3_DOCUMENT

## Interface of the Interface Method

Import Parameter	Description
IT_EXT_ID	Table of key fields displayed on the user interface
IT_INT_ID	Table of existing, additional, internal key fields
IT_DESC	Table for description
IV_TABNAME	Table or view for existence check in the SAP system
Export Parameter	Description
ET_EXT_ID	Table of the key fields that are displayed on the user interface
ET_INT_ID	Table of existing, additional, internal key fields
ET_DESC	Table for description
EV_RETURN	Return parameter (such as BAPIRETURN1)

## Procedure

We recommend that you note the following points during implementation:

- In the SAP system, use the CHECK\_EXISTENCE method of the class CL\_R3\_OLR3\_METHODS in the interface method IF\_OLR3\_PROXY~CHECK\_EXISTENCE for the existence check and the DETERMINE\_DESCRIPTION method of the class CL\_R3\_OLR3\_WORK to determine the language-dependent description.

## Consistency Check

When you create an object link in SAP Portfolio and Project Management using an SAP object type, a consistency check is carried out in the SAP system once a successful existence check has taken place. For this, SAP Portfolio and Project Management calls the method CHECK\_CONSISTENCY of the corresponding SAP Portfolio and Project Management proxy class.

### 3.2.4 Enhancement of the Consistency Check

#### Purpose



This section is aimed at SAP Portfolio and Project Management users with a basic knowledge of the programming environment in the SAP system.

You can enhance the consistency check individually for the following interfaces using the CHECK\_CONSISTENCY method:

- IF\_DPR\_OBL\_PROXY in the DEVELOPMENT\_PROJECTS\_OBJ\_LINK package
- IF\_OLR3\_PROXY in the R3\_OBJECTLINK\_R3 package

See the *Prerequisites* for the Customizing settings you need to make in the Customizing activity *Make Settings for the Linked SAP Objects* to be able to use these individual enhancements.

#### Prerequisites

- You have implemented the interface method IF\_OLR3\_PROXY~CHECK\_CONSISTENCY in the proxy class of an SAP object link in the SAP system.
- You have implemented the interface method IF\_DPR\_OBL\_PROXY~CHECK\_CONSISTENCY in the proxy class of an SAP object link in the SAP Portfolio and Project Management system.
- You have defined a DDIC structure for the object type of an SAP object link for the *Consistency Check* method in the SAP Portfolio and Project Management system.

You have assigned the name of the corresponding proxy class in the SAP system to the object type of an SAP object link.

For more information about these settings, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* in the description of the view *Table/View in SAP System per Object Type and Meth.*

For Portfolio Management object links, see Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object*

*Settings → Define Object Link Types* in the description of the view *Table/View in SAP System per Object Type and Meth.*

## Procedure

We recommend that you note the following points when enhancing the consistency check:

- Use the class builder in the SAP system to create the proxy class for a new SAP object link as a copy of an existing proxy class from the sample implementation in which the consistency check has already been implemented. Then change the implemented interface method `IF_OLR3_PROXY~CHECK_CONSISTENCY` to meet your requirements.
- Do the same in SAP Portfolio and Project Management; copy an existing proxy class from the sample implementation in which the consistency check has already been implemented. Then change the implemented interface method `IF_DPR_OBL_PROXY~CHECK_CONSISTENCY` to meet your requirements.
- Create a new object type for SAP object links as a copy of an existing object type from the sample implementation. For object links from Project Management, copy the Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* and copy all dependent entries. For object links from Portfolio Management, copy the Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*. Afterwards you can change the settings in the lower-level maintenance views.

### 3.2.5 Method `IF_DPR_OBL_PROXY~CHECK_CONSISTENCY`

#### Use

The interface method `IF_DPR_OBL_PROXY~CHECK_CONSISTENCY` has been implemented in the SAP Portfolio and Project Management system in the following proxy classes of the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP Portfolio and Project Management System
Project definition	0PROJECTDEF	CL_DPR_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_DPR_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_DPR_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_DPR_OLR3_DOCUMENT

#### Interface of the Interface Method

Input Parameter	Description
IR_PARENT_OBJECT	Reference of superior SAP Portfolio and Project Management object
IV_OBJECT_KEY	String with the key fields of the identification (if available, see sections 3.1.1 and 3.1.5.7), otherwise with the key fields of the key.
IV_OBJECT_TYPE	Object type

Export Parameter	Description
EV_INCONSISTENT	Boolean indicator; TRUE if inconsistencies are found
EV_ERROR_STRING	String describing the inconsistency

## Procedure

We recommend that you note the following during implementation:

In the SAP Portfolio and Project Management system, use the CHECK\_CONSISTENCY method of the class CL\_DPR\_OLR3\_METHODS in the interface method IF\_DPR\_OBL\_PROXY~CHECK\_CONSISTENCY to call the RFC function module OLR3\_RFC\_CHECK\_CONSISTENCY in the SAP system.

### 3.2.6 Method IF\_OLR3\_PROXY~CHECK\_CONSISTENCY

## Use

The interface method IF\_OLR3\_PROXY~CHECK\_CONSISTENCY has been implemented in the SAP system in the following proxy classes of the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP System
Project definition	0PROJECTDEF	CL_R3_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_R3_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_R3_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_R3_OLR3_DOCUMENT

## Interface of the Interface Method

Input Parameter	Description
IT_INT_ID	Table of the additional, internal key fields
IT_CONSISTENCY	List of active, standard consistency checks
IV_TABNAME	Table or view in the SAP system for determining the object to be checked
Export Parameter	Description
EV_RETURN	Return parameter (such as BAPIRETURN1)

## Procedure

We recommend that you note the following points during implementation:

- Since the coding in these semantic consistency checks cannot be generalized (coding specific to object types or proxy classes), there is no generic CHECK\_CONSISTENCY method in the class CL\_R3\_OLR3\_METHODS.
- You can implement customer-specific consistency checks using the AT\_CONSISTENCY\_CHECK method of the Business Add-In (BAI) OLR3\_CONSISTENCY.

## Consistency Checks in the Standard System

Standard consistency checks are implemented in the ERP system for each SAP object type. You can deactivate individual standard consistency checks in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*:

Object Type	Check	Meaning
0PROJECTDEF	LOCKED	System status <i>Master data lock</i>
	CLOSED	System status <i>Closed</i>
	DELETED	System status <i>Deletion flag</i>
	AUTH_DISP	Display authorization
0DOCUMENT	RELEASED	Document status (of document type) <i>Released</i>
	AUTH_DISP	Display authorization
0RPMSPROJECT	LOCKED	System status <i>Master data lock</i>
	CLOSED	System status <i>Closed</i>
	DELETED	System status <i>Deletion flag</i>
	AUTH_DISP	Display authorization
0RPMDOCUMENT	RELEASED	Document status (of document type) <i>Released</i>
	AUTH_DISP	Display authorization

You **cannot** implement more checks by assigning additional checks in Customizing because consistency checks are non-generalized and semantic verifications. You can deactivate only the existing checks in Customizing. However, you can use the BAI OLR3\_CONSISTENCY to develop other customer-specific checks in the SAP system.

## 3.3 Search Help

If you set the *Search Help* indicator in the object type, the *Search* function is available when you create an object link in SAP Portfolio and Project Management. In this case, SAP Portfolio and Project Management calls the SEARCH\_OBJECTS method of the corresponding SAP Portfolio and Project Management proxy class and transfers the search criteria specified on the user interface.

### 3.3.1 Enhancement of the Search Help

## Purpose



This section is aimed at SAP Portfolio and Project Management users with a basic knowledge of the programming environment in the SAP system.

You can enhance the search help individually for the following interfaces using the SEARCH\_OBJECTS method:

- IF\_DPR\_OBL\_PROXY in the DEVELOPMENT\_PROJECTS\_OBJ\_LINK package
- IF\_OLR3\_PROXY in the R3\_OBJECTLINK\_R3 package

See the *Prerequisites* for the Customizing settings you need to make in the Customizing activity *Make Settings for the Linked SAP Objects* to be able to use these individual enhancements.

## Prerequisites

- You have implemented the interface method IF\_OLR3\_PROXY~SEARCH\_OBJECTS in the proxy class of an SAP object link in the SAP system.
- You have implemented the interface method IF\_DPR\_OBL\_PROXY~SEARCH\_OBJECTS in the proxy class of an SAP object link in the SAP Portfolio and Project Management system.
- You have defined a DDIC structure for the object type of an SAP object link for the *Search Help* method in the SAP Portfolio and Project Management system.

You have assigned the name of the corresponding proxy class in the SAP system to the object type of an SAP object link.

For more information about these settings, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* in the description of the view *Table/View in SAP System per Object Type and Meth..*

For Portfolio Management object links, see Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types* in the description of the view *Table/View in SAP System per Object Type and Meth..*

## Procedure

We recommend that you note the following points when enhancing the search help:

- Use the class builder in the SAP system to create the proxy class for a new SAP object link as a copy of an existing proxy class from the sample implementation in which the search help has already been implemented. Then change the implemented interface method IF\_OLR3\_PROXY~SEARCH\_OBJECTS to meet your requirements.
- Do the same in SAP Portfolio and Project Management; copy an existing proxy class from the sample implementation in which the search help has already been implemented. Then change the implemented interface method IF\_DPR\_OBL\_PROXY~SEARCH\_OBJECTS to meet your requirements.
- Create a new object type for SAP object links as a copy of an existing object type from the sample implementation. For object links from Project Management, copy the Customizing for *SAP Portfolio and Project Management* under *Project Management*

→ *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*, and copy all dependent entries. For object links from Portfolio Management, copy the Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*. Afterwards you can change the settings in the lower-level maintenance views.

- If all search criteria are left blank, the SAP system does not carry out a search. For each search in the SAP system, the system only selects the same number of hits as displayed in SAP Portfolio and Project Management.

### 3.3.2 Method IF\_DPR\_OBL\_PROXY~SEARCH\_OBJECTS

#### Use

The interface method IF\_DPR\_OBL\_PROXY~SEARCH\_OBJECTS has been implemented in the SAP Portfolio and Project Management system in the following proxy classes from the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP Portfolio and Project Management System
Project definition	0PROJECTDEF	CL_DPR_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_DPR_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_DPR_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_DPR_OLR3_DOCUMENT

#### Interface of the Interface Method

Input Parameter	Description
IT_SEARCH_CRITERIA	Search criteria table
IV_OBJECT_TYPE	Object type
IV_MAX_NO_OF_HITS	Maximum number of hits from the search help
Export Parameter	Description
EV_ERROR_STRING	Error string if search fails
ET_SEARCH_RESULT	Hit list

#### Procedure

We recommend that you note the following during implementation:

In the SAP Portfolio and Project Management system, use the SEARCH\_OBJECTS method of the class CL\_DPR\_OLR3\_METHODS in the interface method IF\_DPR\_OBL\_PROXY~SEARCH\_OBJECTS to call the RFC function module OLR3\_RFC\_SEARCH\_OBJECTS in the SAP system.

### 3.3.3 Method IF\_OLR3\_PROXY~SEARCH\_OBJECTS

#### Use

The interface method IF\_OLR3\_PROXY~SEARCH\_OBJECTS has been implemented in the SAP system in the following proxy classes from the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP System
Project definition	0PROJECTDEF	CL_R3_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_R3_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_R3_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_R3_OLR3_DOCUMENT

#### Interface of the Interface Method

Input Parameter	Description
IV_OBJECT_TYPE	Object type
IV_TABNAME	Table or view in the SAP system for determining the object to be checked
IT_SELECT	Table of fields for the select clause
IT_SEARCH_CRITERIA	Table of search criteria used
IV_MAX_ROWS	Maximum number of hits in the search help
Export Parameter	Description
ET_RESULT	Table type for OLR3_R3_TS_SHLP_RESULT
EV_MORE_THAN	Maximum number of hits in the search help reached

#### Procedure

We recommend that you note the following during implementation:

In the SAP system, use the SEARCH\_OBJECTS method of the class CL\_R3\_OLR3\_METHODS in the interface method IF\_OLR3\_PROXY~SEARCH\_OBJECTS for the search help and the method DETERMINE\_DESCRIPTION of the class CL\_R3\_OLR3\_WORK to determine the language-dependent description.

## 3.4 Data Retrieval

The *Evaluation* and *Detail Screen* functions are only available for an existing SAP object link in SAP Portfolio and Project Management if you selected the *Read* indicator in the object

type. In this case, SAP Portfolio and Project Management calls the GET\_DATA method of the appropriate SAP Portfolio and Project Management proxy class.

### 3.4.1 Data Retrieval Enhancement

#### Purpose



This section is aimed at SAP Portfolio and Project Management users with a basic knowledge of the programming environment in the SAP system.

You can enhance data retrieval individually for the following interfaces using the GET\_DATA method:

- IF\_DPR\_OBL\_PROXY in the DEVELOPMENT\_PROJECTS\_OBJ\_LINK package
- IF\_OLR3\_PROXY in the R3\_OBJECTLINK\_R3 package

See the *Prerequisites* for the Customizing settings you need to make in the Customizing activity *Make Settings for the Linked SAP Objects* to be able to use these individual enhancements.

#### Prerequisites

- You have implemented the interface method IF\_OLR3\_PROXY~GET\_DATA in the proxy class of an SAP object link in the SAP system.
- You have implemented the interface method IF\_DPR\_OBL\_PROXY~GET\_DATA in the proxy class of an SAP object link in the SAP Portfolio and Project Management system.
- You have defined a DDIC structure for the object type of an SAP object link for the *data retrieval* method in the SAP Portfolio and Project Management system.

You have assigned the name of the corresponding proxy class in the SAP system to the object type of an SAP object type.

For more information about these settings, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* in the description of the view *Table/View in SAP System per Object Type and Meth.*

For Portfolio Management object links, see Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types* in the description of the view *Table/View in SAP System per Object Type and Meth.*

- You have defined a maximum of nine frames each with a maximum of 100 fields for each object type. For more information about the concrete values in the implementation provided, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*. You define the pool of possible fields in the *Fields per Object Type* view.

#### Procedure

We recommend that you note the following points when enhancing data retrieval:

- Use the class builder in the SAP system to create the proxy class for a new SAP object link as a copy of an existing proxy class from the sample implementation in which data retrieval has already been implemented. Then change the implemented interface method IF\_OLR3\_PROXY~GET\_DATA to meet your requirements.
- Do the same in SAP Portfolio and Project Management; copy an existing proxy class of the sample implementation in which data retrieval has already been implemented. Then change the implemented interface method IF\_DPR\_OBL\_PROXY~GET\_DATA to meet your requirements.
- Create a new object type for SAP object links as a copy of an existing object type from the sample implementation. For object links from Project Management, copy the Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects*, and copy all dependent entries. For object links from Portfolio Management, copy the Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*. Afterwards you can change the settings in the lower-level maintenance views.

### 3.4.2 Method IF\_DPR\_OBL\_PROXY~GET\_DATA

#### Use

The interface method IF\_DPR\_OBL\_PROXY~GET\_DATA has been implemented in the SAP Portfolio and Project Management system in the following proxy classes from the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP Portfolio and Project Management System
Project definition	0PROJECTDEF	CL_DPR_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_DPR_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_DPR_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_DPR_OLR3_DOCUMENT

#### Interface of the Interface Method

Input Parameter	Description
IV_OBJECT_KEY	String with the key fields of the identification (if available, see section 3.1.1 and 3.1.5.7), otherwise with the key fields of the key.
IV_OBJECT_TYPE	Object type
IV_CONTEXT	Context in which the method is called (for example, 'EVALUATION')
Export Parameter	Description
EV_OBJECT_DATA	Contains the XML string, type XSTRING
EV_SYSTEM_ERROR	Type of system error when accessing the object

## Procedure

We recommend that you note the following points during implementation:

- In the SAP Portfolio and Project Management system, use the GET\_DATA method of the class CL\_DPR\_OLR3\_METHODS in the interface method IF\_DPR\_OBL\_PROXY~GET\_DATA to call the RFC function module OLR3\_RFC\_GET\_DATA in the SAP system and to retrieve the data for the *Detail Screen* and *Evaluation* functions.
- You can use the GET\_BUFFER method on the class CL\_DPR\_OLR3\_METHODS if you want to buffer the retrieved data.

### 3.4.3 Method IF\_OLR3\_PROXY~GET\_DATA

#### Use

The interface method IF\_OLR3\_PROXY~GET\_DATA has been implemented in the SAP system in the following proxy classes from the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP System
Project definition	0PROJECTDEF	CL_R3_OLR3_PROJECTDEFINITION
SAP DMS document	0DOCUMENT	CL_R3_OLR3_DOCUMENT
PS Project Definition	0RPMPSPROJECT	CL_R3_OLR3_PROJECTDEFINITION
DMS Document	0RPMDOCUMENT	CL_R3_OLR3_DOCUMENT

#### Interface of the Interface Method

Input Parameter	Description
IV_CONVERT	Indicator for converting field values
IV_LABEL	Indicator for determining field names
IV_LANGU	Language key
IT_INT_ID	Table of fields for internal key
IT_TABNAMES	Tables from which the system has already read generically
IT_DATA	Inbound data container
Export Parameter	Description
ET_DATA	Outbound data container

## Procedure

We recommend that you note the following during implementation:

In the SAP system, use the GET\_DATA method of the class CL\_R3\_OLR3\_METHODS in the interface method IF\_OLR3\_PROXY~GET\_DATA for data retrieval.

## 3.5 Date Retrieval

The system determines a minimum start and a maximum finish date for each object in Project Management using the relevant assigned SAP object links of the object. This date pair is then available for the dates graphic in Project Management and is displayed there as an independent bar.

### 3.5.1 Date Retrieval Enhancement

#### Purpose



This section is aimed at SAP Portfolio and Project Management users with a basic knowledge of the programming environment in the SAP system.

You can individually enhance date retrieval for the IF\_OLR3\_PROXY interface in the package R3\_OBJECTLINK\_R3 using the GET\_DATES method.

See the *Prerequisites* for the Customizing settings you need to make in the Customizing activity *Make Settings for the Linked SAP Objects* to be able to use these individual enhancements.

#### Prerequisites

- You have implemented the interface method IF\_OLR3\_PROXY-GET\_DATES in the proxy class of an SAP object link in the SAP system.
- You have defined a DDIC structure for the object type of an SAP object link for the *date retrieval* method in the SAP Portfolio and Project Management system.

You have assigned the name of the corresponding proxy class to the object type of an SAP object link in the SAP system.

For more information about these settings, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* in the description of the view *Table/View in SAP System per Object Type and Meth.*

#### Procedure

We recommend that you note the following points when enhancing date retrieval:

- Use the class builder in the SAP system to create the proxy class for a new SAP object link as a copy of an existing proxy class from the sample implementation in which date retrieval has already been implemented. Then change the implemented interface method IF\_OLR3\_PROXY~GET\_DATES to meet your requirements.
- Create a new object type for SAP object links as a copy of an existing object type from the sample implementation in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* and copy all dependent entries. Afterwards you can change the settings in the lower-level maintenance views.

### 3.5.2 Method IF\_OLR3\_PROXY~GET\_DATES

#### Use

The interface method IF\_OLR3\_PROXY~GET\_DATES has been implemented in the SAP system in the following proxy classes from the sample implementation:

SAP Object Type	Object Type in SAP Portfolio and Project Management	Name of Proxy Class in the SAP System
Project definition	0PROJECTDEF	CL_R3_OLR3_PROJECTDEFINITION

#### Interface of the Interface Method

Import Parameter	Description
IV_DATE_SET	Date type
IT_WHERE	Table of internal key fields for creating the WHERE condition for database access
Change Parameter	Description
CS_OBL_OBJ_TYPE	<p>Structure with detailed information about the current SAP object link.</p> <p>You transfer the start or finish date as the UTC time stamp in a shortened form (JJJJMMThhmmss) in the following fields:</p> <ul style="list-style-type: none"> <li>AUX_START (Data type: TIMESTAMP)</li> <li>AUX_FINISH (Data type: TIMESTAMP)</li> </ul> <p>Do not change the following fields:</p> <ul style="list-style-type: none"> <li>GUID</li> <li>OBJECT_ID</li> <li>OBJECT_TYPE</li> <li>PROXY_CLASS_R3</li> <li>DESTINATION</li> <li>TABNAME</li> </ul> <p>(Name of the DDIC structure defined in Customizing for the <i>Dates Retrieval</i> method)</p>
Export Parameter	Description
EV_RETURN	Return parameter (Data type: BAPIRETURN)

#### Procedure

We recommend that you note the following points during implementation:

- In the interface method IF\_OLR3\_PROXY~GET\_DATES, use the CONVERT\_EXT\_TO\_WHERE method of the class CL\_R3\_OLR3\_WORK to set up a WHERE table of the data type RSDS\_WHERE\_TAB. Transfer the imported parameters IT\_WHERE and CS\_OBL\_OBJ\_TYPE-TABNAME to the method CL\_R3\_OLR3\_WORK.

- Use the GET\_TIMESTAMP method of the class CL\_R3\_OLR3\_WORK to convert a date of the data type DATS to a UTC time stamp of the data type TIMESTAMP.

## Supported Date Types

The following date types are defined for the SAP object links in the sample implementation for *project definition*:

Proxy Class of the SAP Object Link	Constants of the Class CL_OLR3_WORK	Date Types
CL_R3_OLR3_PROJECTDEFINITION	SC_BASIC	Basic dates
	SC_BASIC_EARLY	Earliest basic date
	SC_BASIC_LATEST	Latest basic date
	SC_FORECAST	Forecast dates
	SC_FORECAST_EARLY	Earliest forecast date
	SC_FORECAST_LATEST	Latest forecast date

## 3.6 Calling an SAP Transaction

You can define settings for creating a service for calling your own SAP transactions.

### Prerequisites

To be able to use SAP transactions for SAP object links with their preconfigured settings, you have to carry out the steps in section 1. Installing an SAP Internet Transaction Server 6.40 for the required SAP system is also a prerequisite. (As of SAP NetWeaver Release 700: Activate the internal SAP Internet Transaction Server (ITS).)

### 3.6.1 Service for an SAP Transaction

A service specifies which transaction is called in the SAP system to display or edit a linked object. If necessary, you can also specify which fields the system automatically fills and which pushbuttons it automatically activates. This means you can skip the initial screen, for example.

### Procedure

Carry out the following steps in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for Calling an SAP Transaction*:

1. Create the service definition in the *Service for an SAP Transaction* view.
2. Define the required parameters in the *Parameters for a Service* view.

This customizing activity can also be reached from *Portfolio Management* → *Base Systems Interfaces* → *Application Object Settings* → *Make Settings for Calling an SAP Transaction*.

Then assign the required object types for object links to the service in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External*

*Systems → Object Links in SAP Systems → Make Settings for the Linked SAP Objects in the Service for an Object Type view.*

For object links from Portfolio Management, assign the required object types for object links to the service in Customizing for *SAP Portfolio and Project Management* under *Portfolio Management → Base System Interfaces → Application Object Settings → Define Object Link Types* in the *Service for an Object Type* view.

## 3.6.2 Service Parameters

### 3.7.2.1 Name of a Parameter in a Start URL

The system uses the parameters to generate a start URL for the SAP GUI for HTML. It then uses this URL to call an SAP transaction via the Internet Transaction Server (ITS). You can tell the system to automatically make the required entries in the initial screen so that the user is directed to the corresponding transaction.

You can enter one or more of the following parameters:

- **Screen field**  
Name of a screen field on the initial screen of an SAP transaction. The name of the screen field must appear on the initial screen of the SAP transaction.
- **~TRANSACTION**  
String for specifying the SAP transaction to be called.
- **~OKCODE**  
String for specifying the OK code of an SAP transaction. The OK code specified in the parameter value is required for skipping the initial screen.
- **~CLIENT**  
String for explicitly specifying the client. The client you specify as the parameter value must exist in the target system. If you do not specify a client, the system determines the client from the RFC destination defined in the object type for object links.
- **~LANGUAGE**  
String for explicitly specifying the language setting. The language abbreviation you enter as the parameter value defines the language in which the system starts the SAP transaction. If you do not enter a language abbreviation, the system uses the current language settings in the SAP Portfolio and Project Management system to call the SAP transaction.

### 3.7.2.2 Value of a Parameter in a Start URL

The Web server address must contain the string *http* or *https*.

Enter the following values in the parameters:

- **Screen field**  
A component of the DDIC structure, which describes the external identification or the internal key of each SAP Portfolio and Project Management object type for object links. If both structures exist, use a component of the identification structure. You can find the names of the DDIC structures in Customizing for *SAP Portfolio and Project Management* under *Define Object Types for Object Links*.

You can determine the components of these data types using transaction SE12 in the ABAP Dictionary.

The name of the screen field must appear on the initial screen of the SAP transaction.

- ~TRANSACTION  
Name of the SAP transaction to be called.
- ~OKCODE  
OK code for the initial screen of the SAP transaction to be called.  
The OK code is required for skipping the initial screen.
- ~CLIENT  
Name of the client in which the SAP transaction is to be started.  
The client must exist in the target system. If you do not specify a client, the system determines the client from the SAP RFC destination defined in the object type for object links.
- ~LANGUAGE  
Language abbreviation for the language in which the system is to start the SAP transaction.  
If you do not enter a language abbreviation, the system uses the current language settings in the SAP Portfolio and Project Management system to call the SAP transaction.

## Dependencies

For Project Management object links, you have to define the Web server address of the Internet Transaction Server in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Structure* → *Define Object Types for Object Links*. This is the Internet Transaction Server to which the SAP system of the transaction to be called is assigned. If the defined Web server address does not contain the *WebGui* service, the system generates the string "scripts/wgate/webgui/!?" in the start URL.

For Portfolio Management object links, you have to define the Web server address of the Internet Transaction Server in Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types*.

For an object, if different services require different web servers, you can define the Web server address per service in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Make Settings for the Linked SAP Objects* in the view *Service for an Object Type*.

For Portfolio Management object links, you can define the Web server address per service in Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Link Types* in the view *Service for an Object Type*.

The Web server defined per service gets preference over the Web server defined for an object as a whole.

## 4 Attributes for Evaluations

We defined the following attributes for the SAP object type *project definition (PS)* in this sample implementation (see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Evaluations* → *Define Attributes and Object Types for Evaluations*):

- Attributes for dates
- Attributes for costs
- Attributes for progress values

Note that:

- Attributes are generally of a semantic nature and can serve as examples for your own evaluations, if necessary. The attributes for dates, costs, and progress values are of a general nature.
- The attribute name in the evaluation must match the field name of the object link type tag.
- In the sample implementation, we did not define any separate severities or threshold values in the evaluation.
- Attributes for evaluation are not applicable for the Portfolio Management object links.

For more information, see the central SAP Portfolio and Project Management documentation on *Evaluations*.

## 5 User-Defined Object Links

You can use the SAP object types from the sample implementation in the following ways:

- Directly
- As a copy
- As a modified copy
- As a semantically similar copy

You can also create your own object type without a template and without using the framework from the sample implementation.

### 5.1 Using the Object Types Delivered As Examples

Carry out the steps from the instructions in the introduction (section 1) to use the SAP object types delivered as examples straight away.

### 5.2 Copying an Object Type from the Sample Development

You use a copied object type if you want to link an SAP object type from two or more SAP systems (or clients of a system) to SAP Portfolio and Project Management using an object link.



You want to create object links in SAP Portfolio and Project Management to WBS elements from the SAP systems ABC client 100 and XYZ client 250.

In this case, you can only use the object type 0WBSELEMENT for one of the SAP systems because the SAP destination must be unique for each object type (see section 1).

Therefore, you have to create a new object type and use the object type 0WBSELEMENT as a template.



The new object type cannot be within the SAP namespace, that is, it cannot begin with a zero.

Copying the entire Customizing settings of an object type takes a long time due to the scattered maintenance of the object type as a secondary object in the evaluation. Therefore, you can use the report RCOPY\_SAP\_OBJECT\_TYPE in the standard system. Start the report using transaction SA38 or SE38 and enter the existing and active SAP template object type as the source and the target object type that does not exist yet as the target. The target object type must be in the customer namespace. The system then copies the entire Customizing settings of an object type in one go.



You copy the SAP object type 0WBSELEMENT (delivered in the sample implementation) to ABC\_WBSELEMENT and XYZ\_WBSELEMENT, and this means you avoid working with the SAP object type completely.

Then, only the steps described in the instructions in the introduction are still necessary for the new object types.

All three object types can be differentiated in the following ways:

- By the logical SAP system you define in Customizing
- By the Web server
- By differing settings in standard Customizing, as necessary

## 5.3 Using an Object Type from the Sample Development in a Modified Form

You want to use an SAP object type from the sample implementation in a modified version. Using the report RCOPY\_SAP\_OBJECT\_TYPE, first copy an SAP object type.

*Using a modified version* means that you change one or more technical settings from section 3 in Customizing. However, the SAP object type must keep its character.

### 5.3.1 Key, Identification, and Description

Copy the DDIC structure used in the standard system for [Key](#) or [Identification](#) and change it as required (see section [3.1.1](#)). Enter your own DDIC structure in Customizing in your own object type.



- Copy the DDIC structure of the description in the same way as the new DDIC name of the key because the DDIC name of the description is derived from the DDIC name of the key (see section 3.1.2).
- Despite changed key fields in the key or identification, the SAP object in the SAP system must still be able to be uniquely defined (see section 3.1.5). Example: You do not use any document parts in the document and you delete the field as the key. Documents with more than one document part can no longer be determined uniquely from SAP Portfolio and Project Management and can therefore no longer be linked.
- All key fields and description fields must still exist in the relevant tables or views you defined in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* → *Table/View in SAP System per Object Type and Meth.*, and *Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Links Types* → *Table/View in SAP System per Object Type and Meth.* (see section 3.1.5).
- Problems that result from changing the DDIC structures for key, identification, or description are not part of the maintenance provided by SAP free of charge.

## 5.3.2 Search Fields

Copy the DDIC structure used in the standard system for search fields and change it as required (see section 3.1.3). Enter the new DDIC structure for your own object type in Customizing for *SAP Portfolio and Project Management*. For more information, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* → *Object Types for Object Links and Portfolio Management* → *Base System Interfaces* → *Application Object Settings* → *Define Object Links Types*.



Problems that result from changing the DDIC structure for the search help are not part of the maintenance provided by SAP free of charge.

### Define an Additional Search Field for Your Own Object Type

If you want to add an additional search field for your own object type by enhancing the DDIC structure for the search fields - OLR3\_TS\_XXX\_SHLP, proceed as follows:

1. Decide whether the existing search help views cover the new search field or whether a new search help view is necessary. For more information about the existing search help views, see Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Connection to External Systems* → *Object Links in SAP Systems* → *Make Settings for the Linked SAP Objects* → *Table/View in SAP System per Object Type and Meth.* in the description of the strategies of method 3.



The new search field must be available in all search help views that exist for the object type or added as required.

2. If this causes a V\_OLR3\_XXX search help view to be changed in the SAP system, do **not** change the search help views V\_OLR3\_XXX in the SAP system, but proceed as follows:
  - a. Copy the relevant view
  - b. Change it
  - c. Add it to your object type in Customizing

## 5.3.3 Proxy Classes

A proxy class can be used by more than one object type. This applies to classes in SAP Portfolio and Project Management and in the SAP system. For example, two object types can use the same SAP Portfolio and Project Management proxy class but different SAP proxy classes.

In the various SAP systems, the BAdI implementations for the consistency check (section 3.3) or data retrieval (section 3.5) can differ without modification when using the proxy classes delivered by SAP.



Carry out your own additional developments in your own proxy classes only. You can copy the proxy class from the template and enter your own proxy class in your own object type.



- Your own proxy class in SAP Portfolio and Project Management must implement the interface IF\_DPR\_OBL\_PROXY.
- Your own proxy class in the SAP system must implement the interface IF\_OLR3\_PROXY.
- Your own proxy classes do not fall under the maintenance provided free of charge by SAP.
- Changes to copied, user-specific classes are not viewed as a modification, however, they do not fall under the maintenance provided free of charge by SAP.

### 5.3.4 Table/View in SAP System per Object Type and Method

Even if you do not change the key fields or search help fields (see section 5.3.1 and 5.3.2), you can change the tables and views of your own object type.



- At least strategy one is required for each method.
- You must be able to uniquely identify the SAP object in new tables or views, too.

## 5.4 Copying an Object Type from the Sample Development and Adapting It to a Semantically Similar SAP Object Type

If you want to use an SAP object type from the sample implementation in a semantically similar way, proceed as follows:

Use the report RCOPY\_SAP\_OBJECT\_TYPE to copy the semantically similar SAP object type.

*Semantically similar* is a term that covers all possible changes to the template object type. Therefore, the information in section 5.3 also applies here. The only difference is that the character of the SAP object type changes in this case.

Due to varying semantics and business environments, it is not possible to specify a procedure that is valid for all possible object types.



- Self-defined, semantically similar object types are not covered by the maintenance provided free of charge by SAP.
- To define a semantically similar SAP object, you require technical and SAP-specific know-how.

## 5.5 Creating Without a Template

Create a new object type for Project Management directly in Customizing for *SAP Portfolio and Project Management* under *Project Management* → *Structure* → *Define Object Types for Object Links*. Create a new object type for Portfolio Management directly in Customizing for *SAP Portfolio and Project Management* under *Portfolio Management* → *Base System*

*Interfaces → Application Object Settings → Define Object Link Types.* Follow the steps outlined in the general documentation there.