



How-To Guide: Transaction Launcher

SAP[®] CRM 7.0

Target Audience

- System administrators
- Technology consultants

Document version: 1.0 – December 2008



SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

© Copyright 2007 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.

SAP Library document classification: PUBLIC

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>

Terms for Included Open Source Software

This SAP software contains also the third party open source software products listed below. Please note that for these third party products the following special terms and conditions shall apply.

1. This software was developed using ANTLR.

2. gSOAP

Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are Copyright (C) 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved.

THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3. SAP License Agreement for STLport

SAP License Agreement for STLPort between

SAP Aktiengesellschaft

Systems, Applications, Products in Data Processing

Neurottstrasse 16

69190 Walldorf, Germany

(hereinafter: SAP)

and

you

(hereinafter: Customer)

a) Subject Matter of the Agreement

A) SAP grants Customer a non-exclusive, non-transferrable, royalty-free license to use the STLport.org C++ library (STLport) and its documentation without fee.

B) By downloading, using, or copying STLport or any portion thereof Customer agrees to abide by the intellectual property laws, and to all of the terms and conditions of this Agreement.

C) The Customer may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrictions.

D) Customer shall maintain the following copyright and permissions notices on STLport sources and its documentation unchanged:
Copyright 2001 SAP AG

E) The Customer may distribute original or modified STLport sources, provided that:

o The conditions indicated in the above permissions notice are met;

o The following copyright notices are retained when present, and conditions provided in accompanying permission notices are met:

Copyright 1994 Hewlett-Packard

Company

Copyright 1996,97 Silicon Graphics

Computer Systems Inc.

Copyright 1997 Moscow Center for

SPARC Technology.

Copyright 1999,2000 Boris Fomitchev

Copyright 2001 SAP AG

Permission to use, copy, modify, distribute and

sell this software and its documentation for

any purposes is hereby granted without fee,

provided that the above copyright notice appear

in all copies and that both that copyright notice

and this permission notice appear in supporting

documentation. Hewlett-Packard Company

makes no representations about the suitability

of this software for any purpose. It is provided

“as is” without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purposes is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Moscow Center for SPARC makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Boris Fomitchev makes no representations about the suitability of this software for any purpose. This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk. Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purposes is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. SAP makes no representations about the suitability of this software for any purpose. It is provided with a

limited warranty and liability as set forth in the License Agreement distributed with this copy. SAP offers this liability and warranty obligations only towards its customers and only referring to its modifications.

b) Support and Maintenance

SAP does not provide software maintenance for the STLport. Software maintenance of the STLport therefore shall be not included.

All other services shall be charged according to the rates for services quoted in the SAP List of Prices and Conditions and shall be subject to a separate contract.

c) Exclusion of warranty

As the STLport is transferred to the Customer on a loan basis and free of charge, SAP cannot guarantee that the STLport is error-free, without material defects or suitable for a specific application under third-party rights. Technical data, sales brochures, advertising text and quality descriptions produced by SAP do not indicate any assurance of particular attributes.

d) Limited Liability

A) Irrespective of the legal reasons, SAP shall only be liable for damage, including unauthorized operation, if this (i) can be compensated under the Product Liability Act or (ii) if caused due to gross negligence or intent by SAP or (iii) if based on the failure of a guaranteed attribute.

B) If SAP is liable for gross negligence or intent caused by employees who are neither agents or managerial employees of SAP, the total liability for such damage and a maximum limit on the scope of any such damage shall depend on the extent to which its occurrence ought to have anticipated by SAP when concluding the contract, due to the circumstances known to it at that point in time representing a typical transfer of the software.

C) In the case of Art. 4.2 above, SAP shall not be liable for indirect damage, consequential damage caused by a defect or lost profit.



D) SAP and the Customer agree that the typical foreseeable extent of damage shall under no circumstances exceed EUR 5,000.

E) The Customer shall take adequate measures for the protection of data and programs, in particular by making backup copies at the minimum intervals recommended by SAP. SAP shall not be liable for the loss of data and its recovery, notwithstanding the other limitations of the present Art. 4 if this loss could have been avoided by observing this obligation.

F) The exclusion or the limitation of claims in accordance with the present Art. 4 includes claims against employees or agents of SAP.

4. Adobe Document Services

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and / or other countries. For information on Third Party software delivered with Adobe document services and Adobe LiveCycle Designer, see SAP Note 854621.

Typographic Conventions

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

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Contents

1 Introduction	8
2 How to Use the Transaction Launcher.....	9
2.1 Prerequisites and Restrictions.....	9
2.2 Integrating an Existing Launch Transaction.....	9
2.3 Creating a New Launch Transaction	9
2.3.1 URL-Based Launch Transaction	9
2.3.2 ITS-Based Launch Transaction	10
2.4 Changing an Existing Launch Transaction.....	10
2.5 Deleting an Existing Launch Transaction	10
2.5.1 Deleting a URL-Based Launch Transaction	10
2.5.2 Deleting an ITS-Based Launch Transaction	11
2.6 Customizing.....	11
2.6.1 Define RFC Destination.....	11
2.6.2 Define Logical System.....	13
2.6.3 Assignment of RFC Destination to Logical System for Synchronous Method Calls..	14
2.6.4 Define Mapped Logical System and URL of ITS	14
2.6.5 Define URLs and Parameters.....	15
2.6.6 Launch Transaction Wizard	17
2.6.7 Copy/Delete Launch Transactions	20
2.7 Additional Information	20
2.7.1 Reverse Proxy.....	20
2.7.2 Avoiding Login Screens.....	21

1 Introduction

The transaction launcher is an integration tool. It enables the display of web pages within the WebClient UI that provide functionality which has not been implemented within WebClient.

There are three different kinds of integration:

- Calling URLs
- Calling a method of the business object repository (BOR) using the SAP GUI for HTML
- Front office (FO) processes in the utilities industry (IS-U)

The UI is either displayed in the whole work area or in a separate browser window. In the latter case, the original WebClient displays a generic work area of the transaction launcher.

Data flow to and from the external UI is supported with some restrictions.

The transaction launcher helps you in case that no other integration of external content is possible. But it is definitely not intended to use WebClient like a "mini portal" and misusing the transaction launcher as main integration tool.

2 How to Use the Transaction Launcher

2.1 Prerequisites and Restrictions

The target page to be displayed must be able to run in an IFRAME, i.e. the page must not try to modify any parent frame (with scripting).

If return data flow from the target page to WebClient is needed, both WebClient and the target page have to run in the same domain. Otherwise cross-site scripting, which is needed for data transport, is not possible due to security restrictions.

Methods of BOR objects can have parameters that are typed to a BOR object (e.g. method EnterReadingDocs of IS-U specific BOR object MTRREADDOC). These object-like parameters are not supported by the transaction launcher.

Note

Do not embed the UI component of the transaction launcher into another UI component as this can cause losing the content of the IFRAME.

For more information about the SAP GUI for HTML, see [SAP Help Portal](#).

2.2 Integrating an Existing Launch Transaction

Launch transactions can be integrated into the navigation bar. To get an overview of existing launch transactions, display the view cluster CRMV_IC_LTX_ID, e.g. with transaction SM34. Remember the launch transaction ID that you want to call.

In transaction CRMC_UI_NBLINKS, you need a logical link of type C (Launch Transaction) with the target ID EXECLTX. As parameter class use CL_CRM_UI_LTX_NAVBAR_PARAM and as parameter use the launch transaction ID.

Assign the logical link to either a work center or a work center link or to a direct link that is part of your navigation bar profile. You can also define the logical link as target in the generic outbound plug mapping, which is responsible of cross-component navigation.

2.3 Creating a New Launch Transaction

2.3.1 URL-Based Launch Transaction

To create a URL-based launch transaction, first define the URL that you want to call. You can define URLs in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Define URLs and Parameters*.

Dependent on a URL_ID, you specify different settings of a URL. URLs can either call an arbitrary web page or a BSP application.

As a second step, include the URL_ID as one parameter of your launch transaction using the launch transaction wizard. You can find the launch transaction wizard in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Configure Transaction Launcher*.

After creating the new launch transaction, you can integrate it into the navigation bar. For more information, see *Integrating an Existing Launch Transaction* [page 9].

—

2.3.2 ITS-Based Launch Transaction

ITS-based launch transactions are launch transactions that either call a BOR object or a front office process. In both cases, the UI is built by the SAP GUI for HTML, which you can call via the service WEBGUI of the internet transaction server (ITS).

ITS-based launch transactions use the so-called mapped logical system, which specifies the remote system. Thus, they require certain settings in Customizing. You can find information on how to maintain Customizing in the following sections:

- *Define RFC Destination* [page 11]
- *Define Logical System* [page 13]
- *Assignment of RFC Destination to Logical System for Synchronous Method Calls* [page 14]
- *Define Mapped Logical System and URL of ITS* [page 14]

Use the mapped logical system within your launch transaction with the launch transaction wizard. You can find the launch transaction wizard in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Configure Transaction Launcher*.

After creating the new launch transaction, you can integrate it into the navigation bar. For more information, see *Integrating an Existing Launch Transaction* [page 9].

2.4 Changing an Existing Launch Transaction

You can change an existing launch transaction using the launch transaction wizard. You can find the launch transaction wizard in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Configure Transaction Launcher*. You can change some (but not all) values. In particular, you can adjust the parameters of the data flow.

If you intend to change a value that is not changeable in the launch transaction wizard, you have to create a new launch transaction that uses a different handler class. This is due to the fact that settings within the wizard influence the generated coding of the handler class.

2.5 Deleting an Existing Launch Transaction

2.5.1 Deleting a URL-Based Launch Transaction

You can delete existing launch transactions in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Copy/Delete Launch Transactions*.

1. Locate the entry that is to be deleted and view its details.
2. Verify that the transaction type is B (URL Transaction) and get the value from the *Definition 1* field. This is the URL_ID.
3. To check whether your URL_ID is used only within your launch transaction, enter the URL_ID as search criteria in the *Definition 1* field (e.g. with transaction SE16) in table CRMC_IC_LTX_ID.
4. If you get only one result entry, delete the entry of your URL_ID including all dependent entries in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Define URLs and Parameters* and save.

5. Get the name of the handler class in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Copy/Delete Launch Transactions*.
6. Delete the class in transaction SE24.
7. Delete your entry including all dependent entries in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Copy/Delete Launch Transactions* and save.

2.5.2 Deleting an ITS-Based Launch Transaction

You can delete existing launch transactions in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Copy/Delete Launch Transactions*.

1. Locate the entry that is to be deleted and view its details.
2. Verify that the transaction type is either A (BOR Transaction) or C (FO Transaction) and get the name of the handler class.
3. Delete the class in transaction SE24.
4. Delete your entry including all dependent entries in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Copy/Delete Launch Transactions* and save.

2.6 Customizing

2.6.1 Define RFC Destination

RFC destinations represent technical connections to remote systems, e.g. from SAP CRM to a backend ERP system. The ID of an RFC destination to an SAP system normally consists of <system_id>CLNT<client>, e.g. QWCCCLNT555.

You can define RFC destinations using transaction SM59.

1. Select an RFC destination of type 3 (ABAP Connection).
2. Specify a description.
3. Enter necessary technical settings.

RFC Destination	QWCLNT555		
Connection Type	3	ABAP Connection	Description
Description			
Description 1	QWCLNT555 - ABAP Connection to ERP Backend Client		
Description 2			
Description 3			
Administration Technical Settings Logon & Security MDMP & Unicode Special Options			
Target System Settings			
Load Balancing Status			
Load Balancing	<input checked="" type="radio"/> Yes	<input type="radio"/> No	
Target System	QWC		
Msg. Server	us4100.wdf.sap.corp		
Group	PUBLIC		
Save to Database as			
Save as	<input checked="" type="radio"/> Hostname	<input type="radio"/> IP Address	us4100.wdf.sap.coi
Gateway Options			
Gateway Host			Delete
Gateway service			

 Note

You can verify the correctness of the technical settings by executing a connection test.

4. Enter necessary logon information.

RFC Destination	QWCCCLNT555	
Connection Type	3	ABAP Connection
Description		
Description 1	QWCCCLNT555 - ABAP Connection to ERP Backend Client	
Description 2		
Description 3		
Administration Technical Settings Logon & Security MDMP & Unicode Special Options		
Security Options		
Trusted System/Logon Screen Status		
Trusted System	<input checked="" type="radio"/> No	<input type="radio"/> Yes
	<input type="checkbox"/> Logon Screen	
Status of Secure Protocol		
	<input checked="" type="radio"/> Inactive	<input type="radio"/> Active
Authorization for Destination		
Logon		
Language		
Client	555	
User		<input checked="" type="checkbox"/> Current User
PW Status	is initial	

 Note

You can verify the correctness of the logon settings by executing a remote logon. This only works if the user is a dialog user.

In case that the remote system trusts the local system you can avoid an authentication UI during the execution of your launch transaction.

2.6.2 Define Logical System

This is cross-client Customizing. You can define logical systems via one of the following options:

- Customizing for *SAP NetWeaver* under *Application Server* → *IDoc Interface / Application Link Enabling (ALE)* → *Basic Settings* → *Logical Systems* → *Define Logical System*
- Transaction BD54
- Maintenance view V_TBDLS

A logical system always represents the combination of an SAP system ID and one of its clients. Therefore stick to the naming convention <system_id>CLNT<client>, e.g. QWCCCLNT555.

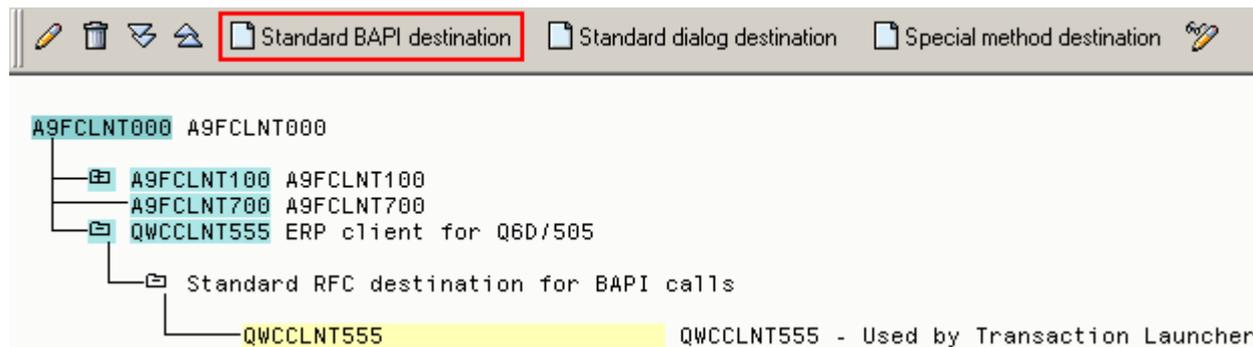
1. Access the target system/client and call transaction SCC4.
2. Get the logical system defined in this transaction.
3. Verify in your CRM system that there is an entry in transaction BD54 with exactly the same logical system.

There is always an entry that represents your own system and client, i.e. <own_system_id>CLNT<own_client>, which corresponds to the settings in transaction SCC4 of your own system and client.

2.6.3 Assignment of RFC Destination to Logical System for Synchronous Method Calls

If you only want to call into your own logical system again you do not need any settings introduced in this section.

To call into a remote system, assign a standard RFC destination for BAPI calls to the logical system that represents your remote system in transaction BD97. Otherwise it is not possible to determine the technical settings of an RFC destination that correspond to the logical system.



2.6.4 Define Mapped Logical System and URL of ITS

In the maintenance view CRMV_IC_BORADM, you can define a mapped logical system and assign a logical system to it. Additionally, you define the URL of the ITS here.

You can also use transaction CRMS_IC_CROSS_SYS (Transaction Launcher Logical Systems) to define a mapped logical system.

A mapped logical system allows you to use a higher level of abstraction. For example, you can define a mapped logical system `ERP` that represents your `ERP` backend system. This mapped logical system is used in multiple launch transactions. After e.g. a transport from your test landscape to your productive landscape, you only have to update the logical system in the maintenance view CRMV_IC_BORADM, which can also be accessed via transaction CRMS_IC_CROSS_SYS. All launch transactions stay untouched as they use the mapped logical system.

SAP recommends to define a mapped logical system `OWNLOGSYS` that uses the logical system of your own system/client. For more information, see *Define Logical System* [page 13].

Additionally, specify the settings of the ITS of the remote system:

- Maintain the client that you used in the RFC destination and that is assigned to the logical system.
- Maintain the URL that calls the service WEBGUI of the ITS in the remote system. The URL has the format
`<protocol>://<host>:<port>/<path>/!?!~transaction=<transaction>&~okcode=ICEXECUTE&~disconnectonclose=1.`

2.6.4.1 How to Get the Different Parts

- `<protocol>`: Clarify whether you want to use HTTP or HTTPS.
- `<host>`: Call transaction SMICM in the remote system/client. In the menu, choose *Goto > Services*, select the line that corresponds to your protocol, and display the details. Here you get the parts `<host>` and `<port>`.

- `<port>`: See `<host>`.
- `<path>`: Call transaction SICF in the remote system/client. Search for the service name WEBGUI. In case of multiple results, prefer the path containing `gui`. Most probably service WEBGUI exists at path `sap/bc/gui/sap/its/webgui`.
- `<transaction>`: There are three possible values depending on the type of remote system:
 - IC_LTX: Use this transaction when calling the ITS of your own CRM system.
 - IC_LTXE: Use this transaction when calling the ITS of an ERP system of at least release 2005 (SAP_ABA 700 or higher).
 - IC_LTXR: Use this transaction when calling the ITS of an R/3 system with PI release track 2004_1_46C to 2004_1_500.

 Note

If the corresponding transaction (and a function group with the same name as the transaction) is missing in your remote system, refer to SAP Note 990216.

Any URL of the ITS containing `*RABOX*` is outdated and not supported any longer. For more information, see SAP Note 990216.

2.6.4.2 Dynamic Determination of Logical System

The settings above allow the static determination of a remote system based on customizing of the launch transaction and the mapped logical system. In some cases, this is not sufficient, e.g. when invoices of different remote systems need to be displayed.

Therefore, a dynamic determination is needed.

There are two possibilities:

- The transaction launcher gets an entity of type `ICBORWrapper` with the data collection of the navigation. The entity represents the BOR object of the remote system and contains the correct logical system of the BOR object. This logical system is used on displaying BOR objects. (Available since WEBCUIF 700 EhP1.)
- The transaction launcher checks the global data context (GDC) for the parameter `LOGICAL_SYSTEM`. This parameter is also used on creating BOR objects. This option needs an activated indicator `multiple systems` in the customizing of mapped logical systems. (Available since WEBCUIF 700.)

The different sources of logical systems have different priorities (lowest number means highest priority):

1. Coding in the launch transaction handler class
2. Attribute `LOGSYS` of an existing `ICBORWrapper`
3. Parameter `LOGICAL_SYSTEM` of the global data context
4. Static determination based on customizing of the launch transaction and the mapped logical system

2.6.5 Define URLs and Parameters

For launch transactions of type B (URL Transaction), you have to define details of the URL request. You can define the details in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Define URLs and Parameters*.

When defining URLs, you can edit the following settings:

Setting	Action
<i>Request Method</i>	Define whether the request that calls the URL uses HTTP method GET or POST. By default, HTTP method GET is used.
<i>HTTPS</i>	Set this indicator if you prefer using HTTPS instead of HTTP for secure data transfer.
<i>Hide Dialog</i>	Set this indicator if the page of the URL only displays data (in contrast to a maintenance page). This suppresses a data loss dialog when closing or leaving the page.
BSP URL (BSP application)	Use these settings for calling a BSP application.
<i>Mapped LogSys</i>	The mapped logical system specifies your remote system. For more information, see <i>Define Mapped Logical System and URL of ITS</i> [page 14].
<i>Application</i>	Enter the BSP application name.
<i>Page</i>	Enter the name of the start page of the BSP application.
Non-BSP URL	Use these settings for calling a plain URL.
<i>Mapped LogSys</i>	The mapped logical system specifies your remote system. You only need this parameter when you want to call a URL that points to an object contained in an SAP system. For more information, see <i>Define Mapped Logical System and URL of ITS</i> [page 14].
<i>Determine Host/Port</i>	The transaction launcher can determine host and port of the remote system. You only need this parameter when you want to call a URL that points to an object located in an SAP system. If you have a reverse proxy within your CRM system landscape, also use this indicator.
<i>URL / URL Continued</i>	Either specify a complete URL including host, port, path, and file (e. g. <code>http://www.google.com</code>) or define the path and file that is about to be called only (e. g. <code>/sap/bc/webdynpro/sap/wdt_flightlist</code>) by editing <i>Mapped LogSys</i> and <i>Determine Host/Path</i> as well. If you define the path and file only, the transaction launcher dynamically combines the different parts of the URL.
<i>Portal Integration</i>	At runtime of the WebClient, the transaction launcher checks whether it runs within an SAP NetWeaver Portal. In that case, the transaction launcher calls the object-based navigation (OBN) of the portal instead of calling any URL. The parameters below define the target of the OBN call.  Note Only if the WebClient runs within an SAP NetWeaver Portal, the settings of portal integration are evaluated. Otherwise, they are ignored and the settings above are used.

Setting		Action
	<i>System Alias</i>	All business objects known in the portal have a system alias. To find out the necessary system alias, look at the business object in the portal.
	<i>Object Name</i>	Enter the name of the business object within the content repository of the portal.
	<i>Portal Operation</i>	Specify whether the business object should be e.g. displayed or edited.
	<i>Combine OBN Parameters</i>	<p>The parameters System Alias, Object Name, and Portal Operation each have exactly one corresponding counterpart in the interface of the OBN call. However, all application specific parameters go into only one parameter of the OBN call.</p> <p>The indicator specifies whether application specific parameters are just concatenated name value pairs (param1=value1&param2=value2) or the concatenated and URL-encoded value of the parameter <code>DynamicParameter=URLEncoded</code> (param1=value1&param2=value2). To use the latter format, set the indicator.</p>

Each URL can have multiple parameters assigned to it. You can assign these parameters in the sub view *Define Parameters*.

In this sub view, you only define the parameter names in the *Description* field. The corresponding parameter values are specified in the launch transaction wizard. The wizard only offers the parameters defined in this sub view. It is not possible to define additional parameters within the wizard.

There are different element types. The element type *Import* means that this parameter is to be passed from WebClient to the target application.

2.6.6 Launch Transaction Wizard

You open the launch transaction wizard in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Configure Transaction Launcher*.

The launch transaction wizard asks for different settings of a launch transaction step by step. It is possible to either create a new launch transaction or to change some settings of an existing launch transaction.

After finishing the steps for applying settings, the wizard generates a launch transaction handler class and creates or changes an entry in the customizing.

For more information on the settings in the customizing, see *Copy/Delete Launch Transactions* [page 20].

2.6.6.1 Entries

Each launch transaction is identified by a unique launch transaction ID. To change an existing launch transaction, use the value help. Otherwise, directly enter a new ID.

If you do not need to pass any data to the target application, leave the *Component Set* field empty. If you need parameters to be filled with data, enter the GenIL component set that fits to the required data.

2.6.6.2 Technical Details

Enter a description of the launch transaction that explains which target application is called.

The class name is used to generate the launch transaction handler class. For more information, see *Result* [page 19]).

If you want the launch transaction to be executed in a new browser window, activate the indicator *New Window* (before WEBCUIF700 this indicator was called *Stateful*).

In the interaction center environment, you can force the interaction center agent to close open launch transactions before ending the interaction. Therefore, an activated indicator *raise veto* causes a check on ending an interaction whether there are still unclosed launch transactions. If yes, then the interaction handling will be interrupted.

With the *Source Session Type* drop down list box, you can specify that the current session ID is passed to the remote part of the launch transaction at runtime. There, this session information can be used for callbacks. This feature was introduced with WEBCUIF700 EhP1.

2.6.6.3 Further Technical Details

The *Object Type* field and *Object Action* field are not used during the processing of the launch transaction. You can leave them empty.

Specify the launch transaction type for a launch transaction ID you have created on the *Entries* screen. If you have selected an existing launch transaction ID, you cannot change the value in the *Transaction Type* field. Launch transaction types that are not listed here are obsolete. Possible values are:

- A BOR Transaction
- B URL Transaction
- C Front Office (FO) Transaction

Depending on the chosen launch transaction type, the *Details* section offers different options.

BOR Transaction

For BOR transactions, the following fields are shown in the *Details* section:

- Logical System: Points to the target system which contains the BOR object that is to be called
- BOR Object Type: Provides the technical name of the business object to be called
- Method Name: Shows the method of the BOR object that is to be executed

Value helps for *BOR Object Type* and *Method Name* are provided as soon as the mapped logical system has been specified.

Ensure that the BOR object is available in the target system. You can access the BOR objects with transaction SWO1. BOR objects must at least have the status *Implemented*. The indicator *Synchronous* must be set within the properties of the BOR method.

URL Transaction

URL transactions refer to URL Ids. For more information, see *Define URLs and Parameters* [page 15].

Front Office Transaction

Front office transactions refer to the ID of a front office process in a target system that is specified in the *Logical System* field. Front office processes only exist in systems of the utilities industry (IS-U).

2.6.6.4 Transaction Parameters

In this step, you can define the values of parameters that are to be transferred to the called transaction (i.e. import to the launch transaction).

There are import parameters in the interface of BOR methods. You can see them via transaction SWO1: Display your BOR object type, set the cursor on a method and choose *Parameters*. These import parameters are automatically provided in the step *Transaction Parameters*.

 Note

Parameters that are manually added to a BOR object type are excluded from this list as the transaction launcher cannot handle them.

URL parameters are provided according to the settings in *Define URLs and Parameters* [page 15].

Parameters can be a simple name value pair. In this case, you can use the *Value* field to define the value. When parameters consist of a structure, the table below the *Value* field provides the individual fields of the structure.

Values can be derived from different sources:

- Direct input of the value as literal
- System table SYST: You can use the syntax `%<field_name>%`, e.g. `%UNAME%`. It is possible to use the value help and select *System*. Select a corresponding entry in the table below the list.
- Global data context (GDC): The GDC has multiple entries. These entries can be filled during the runtime of WebClient with an entity of the BOL. In the value help, it is possible to select one of these GDC entries, e.g. `datacontextCURRENTBT`. You can expand the entity according to the underlying data model.

 Note

In case you have specified a GenIL component set in step *Entries* [page 17], you only get the GDC entries that fit to the specified component set.

Double-click an object of the model and select one of the attributes in the *Attributes* table below the list of component sets.

- Activity clipboard: This is very similar to the global data context described above. The activity clipboard is also mainly used within the interaction center.
- You can pass an entity during the runtime of WebClient to the transaction launcher component. This entity has to be part of the data collection of the navigation. Within the generated handler class, you can access this navigation collection again in its method `PREPARE_DATA_FLOW` and fill the parameter value programmatically.

2.6.6.5 Activity Clipboard Integration

The transaction launcher supports only one return parameter. This parameter represents the processed object.

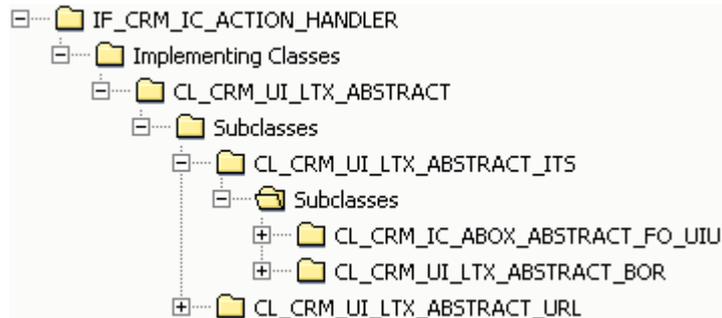
Within this step, you can define whether the processed object is put to the activity clipboard or not. This is mainly of interest in the interaction center environment.

2.6.6.6 Result

The launch transaction wizard puts the entered information into both customizing tables and generated coding. Code and customizing have to stay in sync all the time. Therefore, make changes on the parameters only with the launch transaction wizard.

For more information on customizing, see *Copy/Delete Launch Transactions* [page 20].

The launch transaction handler class is generated by the launch transaction wizard. In any case, it inherits from class `CL_CRM_UI_LTX_ABSTRACT`, which implements the interface `IF_CRM_IC_ACTION_HANDLER`. This interface provides the most important methods for executing the launch transaction. Depending on the launch transaction type, the generated class additionally inherits from one or two additional classes:



The generated class is initially put into the local objects (package `$TMP`) of the current user. Therefore, it is not transported to any successive system. In case the corresponding launch transaction customizing is transported, the handler class will be generated on the fly on executing the launch transaction.

You might want to influence the standard behavior of the launch transaction by your own coding. You can therefore change or redefine the methods of the handler class. In this case, you have to assign the class to a normal package and transport it to all successive systems.

Caution

A complete rerun of the wizard will destroy your coding.

2.6.7 Copy/Delete Launch Transactions

You can copy or delete launch transactions in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Copy/Delete Launch Transactions*.

You can also copy or delete launch transaction in the view cluster `CRMV_IC_LTX_ID`.

Most of the fields on the *Details* screen can also be found in the launch transaction wizard. The following fields are additional fields:

- *Version*: The version is incremented with any run of the launch transaction wizard.
- *Definition 1-4*: These fields contain different data depending on the launch transaction type. You can see the correct labels in the wizard.

2.7 Additional Information

2.7.1 Reverse Proxy

A reverse proxy is located between the browser, which is in the internet, and the CRM web application server, which is in the intranet.

The browser never knows the address of the CRM server and has no direct access to it. All communication has to be done via the reverse proxy.

Therefore, any URL which is sent to the browser in order to be called has to point to the reverse proxy.

The transaction launcher generates URLs containing the host name of the reverse proxy for the following:

- BSP application
- URL that has the indicator *Determine Host and Port*

For more information, see [Define URLs and Parameters \[page 15\]](#).

In case these two types of URLs are located in the source system of the launch transaction, the generated URL contains the host address of the reverse proxy that has called the WebClient.

In case these two types of URLs are located in a remote system, the generated URL is built according to the entries of table HTTPURLLOC of the remote system. For more information on the table HTTPURLLOC, see [SAP Help Portal](#).

You can find more general information about remote proxy scenarios in the SAP Developer Network Wiki at [Using Proxies](#).

2.7.2 Avoiding Login Screens

Calling into a system via HTTP/HTTPS requires a user authentication. There are three ways of avoiding a separate login screen:

- User and password as URL parameter
- Single Sign On (SSO)
- Personal certificates

2.7.2.1 User and Password as URL Parameter

This is a highly insecure way of providing a user authentication. You can use this for testing purposes with a pure display user.



Never use this technique in a productive environment.

Append the parameters *sap-user* and *sap-password* to the URL which calls into the remote system.

- BOR-based launch transactions: Adjust the URL in the maintenance view CRMV_IC_BORADM or transaction CRMS_IC_CROSS_SYS. For more information, see [Define Mapped Logical System and URL of ITS \[page 14\]](#).
- URL-based launch transactions: Adjust the URL in Customizing for *Customer Relationship Management* under *UI Framework* → *Technical Role Definition* → *Transaction Launcher* → *Define URLs and Parameters*. For more information, see [Define URLs and Parameters \[page 15\]](#).

2.7.2.2 Single Sign On (SSO)

First Request with Authentication

During the login to the WebClient UI, you already provide your user authentication. Depending on the system parameter `login/create_sso2_ticket` (its value should be 2), the server creates the cookie `MYSAPSSO2`. The server uses its private key for encrypting the user name and puts the result into the cookie. The cookie gets a validity range consisting of domain and path (e.g. `domain=wdf.sap.corp` and `path=/`) and it is sent back to the browser.

Later Requests

If the browser calls a URL within the validity range of the cookie it will automatically attach the cookie to the request.

Depending on the system parameter `login/accept_sso2_ticket` (its value should be 1), the receiving server looks into the cookie and gets the issuer of the cookie. If the server trusts the issuer (see transaction STRUST) it will decrypt the user name with the public key of the issuer and use the user name. Due to the trusting, it is not necessary to bring up a login page.



The user name must be the same in both the source and destination system.

2.7.2.3 Personal Certificates

You can store certificates in the browser. Depending on the login of the operating system, the browser automatically attaches the correct personal certificate to any HTTPS request. The certificate contains the identity of the user. Within the SAP system, there must be a mapping of the identity used in the certificate to the user name.

As this only works with HTTPS, it is rarely used.