CRM Interaction Center Consultant’s Cookbook – “Maximizing User Experience”

Valid for CRM 7.0 EHP1 and above
Copyright

© Copyright 2012 SAP AG or an SAP affiliate company. 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.

National product specifications may vary.

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 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. Please see http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark for additional trademark information and notices.

SAP CRM “How-to” Guides and Consultant Cookbooks are intended to simplify the product implementation and development of additional functionality that is not part of the standard delivery. This document shall serve as an example for consultants to apply to their specific situation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP CRM. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer:

The information in this document is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. This presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation and SAP’s strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information on this document is not a commitment, promise or legal obligation to deliver any material, code or functionality. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This document is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.
Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔴</td>
<td>Caution</td>
</tr>
<tr>
<td>📜</td>
<td>Example</td>
</tr>
<tr>
<td>🔭</td>
<td>Note</td>
</tr>
<tr>
<td>🌟</td>
<td>Recommendation</td>
</tr>
<tr>
<td>📊</td>
<td>Syntax</td>
</tr>
</tbody>
</table>

Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example text</td>
<td>Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options.</td>
</tr>
<tr>
<td></td>
<td>Cross-references to other documentation.</td>
</tr>
<tr>
<td>Example text</td>
<td>Emphasized words or phrases in body text, titles of graphics and tables.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td>Example text</td>
<td>Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Keys on the keyboard, for example, function keys (such as F2) or the ENTER key.</td>
</tr>
<tr>
<td>Example text</td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td>&lt;Example text&gt;</td>
<td>Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.</td>
</tr>
</tbody>
</table>
Content

1.1 Introduction ............................................................................................................. 6
  1.1.1 Purpose and Audience ......................................................................................... 6
  1.1.2 Prerequisites .......................................................................................................... 7
  1.1.3 Release Relevance and SAP Notes ......................................................................... 7
1.2 Guiding Principles ..................................................................................................... 8
1.3 Processes and Features ............................................................................................. 8
1.4 Basic Setup .................................................................................................................. 9
  1.4.1 CRM-ECC Connection .......................................................................................... 9
  1.4.2 Customizing .......................................................................................................... 10
1.5 Example Solution Realization ................................................................................... 10
  1.5.1 Preparation ............................................................................................................ 10
    1.5.1.1 Technical Requirements .................................................................................. 10
    1.5.1.2 UI Screens ...................................................................................................... 10
    1.5.1.3 UI Components .............................................................................................. 11
  1.5.2 Solution Mock-up ................................................................................................. 14
  1.5.3 High Level Technical Rationale ........................................................................... 14
  1.5.4 Step_by_Step Deployment of the Example Solution with Code
      Examples .................................................................................................................. 14
      1.5.4.1 Creation of an overview page with a tile layout .......................................... 14
      1.5.4.2 Creation of component usages ...................................................................... 21
      1.5.4.3 Add standard windows to master component
          ZICCOOKBOOK ................................................................................................. 25
      1.5.4.4 Create Component usage for the sales order
          items ....................................................................................................................... 31
      1.5.4.5 Make all component usages available to the
          overview page ......................................................................................................... 33
      1.5.4.6 Redefine methods ........................................................................................ 35
      1.5.4.7 Configure Navigation Bar Profile ................................................................. 41
  1.5.5 Solution Results .................................................................................................... 44
1.6 Improving call efficiency by using UI Shortcuts and Intent Driven
    Interactions .................................................................................................................. 45
  1.6.1 UI Shortcuts .......................................................................................................... 45
  1.6.2 Intent Driven Interactions ..................................................................................... 47
1.7 Appendix ...................................................................................................................... 54
  1.7.1 Helpful Links ........................................................................................................ 54
  1.7.2 CRM Performance Analysis and Best Practices .................................................... 54
    1.7.2.1 Preparation and Purpose ................................................................................. 54
    1.7.2.2 Performance Analysis ..................................................................................... 54
  Network and client PC .................................................................................................... 56
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client PC and browser</td>
<td>56</td>
</tr>
<tr>
<td>Compression</td>
<td>57</td>
</tr>
<tr>
<td>Profile Parameter</td>
<td>57</td>
</tr>
<tr>
<td>ICF and BSP compression</td>
<td>57</td>
</tr>
<tr>
<td>SMICM =&gt; Internet connection manager</td>
<td>59</td>
</tr>
<tr>
<td>SAP Web Dispatcher</td>
<td>60</td>
</tr>
<tr>
<td>NetWeaver AccAD</td>
<td>60</td>
</tr>
<tr>
<td>ABAP Performance Analysis</td>
<td>60</td>
</tr>
<tr>
<td>ST02, ST06, ST13</td>
<td>60</td>
</tr>
<tr>
<td>Transactions SE30/SICF/SAT</td>
<td>63</td>
</tr>
<tr>
<td>DB Performance Analysis</td>
<td>65</td>
</tr>
<tr>
<td>RFC Performance analysis</td>
<td>66</td>
</tr>
<tr>
<td>1.7.2.3 Middleware</td>
<td>67</td>
</tr>
<tr>
<td>1.7.2.4 Housekeeping</td>
<td>75</td>
</tr>
<tr>
<td>Org-Management buffering</td>
<td>75</td>
</tr>
<tr>
<td>Application Log Size (BALHDR)</td>
<td>75</td>
</tr>
<tr>
<td>Middleware tables SMW<em>_BDOC</em> can grow very large</td>
<td>75</td>
</tr>
<tr>
<td>MIME Repository consistency check</td>
<td>75</td>
</tr>
<tr>
<td>1.7.2.5 Performance tips and tricks</td>
<td>76</td>
</tr>
<tr>
<td>1.7.2.6 Enqueue Server</td>
<td>76</td>
</tr>
<tr>
<td>Security enhancement available in CRM</td>
<td>76</td>
</tr>
</tbody>
</table>
1.1 Introduction

1.1.1 Purpose and Audience

This cookbook explains how to combine multiple UI components in one view, provides configuration tricks, and steps on how to analyze system performance, so customers that deploy the SAP CRM Interaction Center WebClient can make use of these resources in a way which will achieve higher performance and usability rates; decrease the call time, and provide better customer service.

This document is a collection of guiding principles, hints, and tips which have proven to be valuable and serves as an overview of possibilities. This document complements standard documentation and is not part of the standard product delivery. It is purely a consulting solution approach that can be adjusted to some product.

This cookbook is aimed at members of CRM WebClient UI implementation teams, for example, consultants, project leads and technical resources. It supplements the SAP CRM 7.0 standard documentation.

As of SAP CRM Release 2006s, the IC WebClient shares the UI Framework and it is a part of the generic CRM WebClient UI. Therefore, you may also consult the SAP CRM documentation for your IC WebClient projects.

We also recommend the following documents:

SAP CRM – Installation and Upgrade Information: service.sap.com/crm-inst

- Master Guide
- Security Guide
- Upgrade Guide
- Installation Guides
- Solution Management Guide
- Configuration Guides for all released scenarios

Please see the appendix for more helpful links.

The following best practice configuration guides can be used to configure the scenarios described in this cookbook:

- **B09: CRM Customizing Replication**
- **C01: CRM Organizational Model**
- **C03: CRM Master Data Replication**
- **C04: CRM WebClient UI - Basic Settings**
- **C07: CRM WebClient UI for Sales**
1.1.2 Prerequisites

1.1.3 Release Relevance and SAP Notes

This cookbook is relevant to CRM 7.0 EHP1 and above.

Recommended screen resolution: 1280 x 1024

Before you begin the installation and configuration of SAP CRM Interaction Center, please make sure that the system has the most up-to-date version of the relevant SAP Notes. These SAP Notes contain important information for installation, configuration, and corrections concerning CRM Interaction Center WebClient.

1.2 Guiding Principles

In order to achieve efficiencies in the interaction center, companies look for ways to decrease the total time per call and still provide excellent customer service. Combining multiple components in one page is an excellent way to achieve this.

The following guiding principles were followed when putting together this cookbook:

- The solution provided needed to support the following improvements in the interaction center:
  - Number of navigation steps can be reduced
  - The space on the screen can be optimized
  - Contained units of functionality can be optimized by minimizing the number of clicks it takes to execute the particular functionality. For example: Too many clicks to perform a product search
  - Mouse moving time can be reduced by minimizing the amount of navigational links that need to be clicked

- The example code provided supports the following:
  - Reuse standard components as much as possible
  - Be generic enough, so the logic can be reused by SAP customers deploying the same functionality in other scenarios
  - No customization to the standard SAP code is provided
  - Enhancements are to be performed in the customer name space

1.3 Processes and Features

The enhancements depicted in this cookbook support the following business processes:

Interaction Center WebClient – Sales for outbound Call scenario

1. Agent opens the call list assigned to him/her (IC default page)
2. Agent selects a call from the call list and starts the outbound call to the customer
   a. When the outbound call is placed, business partner details are available. System automatically confirms the customer and contact person
3. Agent can see relevant customer information
4. Agent sees product proposal directly in the CRM Sales order as line items. The proposal is generated based on the confirmed customer’s current products purchased on previous sales orders
5. Agent adds the appropriate quantities in the sales order and delete those line items that are not requested by the customer
6. Agent confirms sales order and clicks on ‘Submit’
7. Sales order submitted is assigned to the current interaction record automatically and it’s available via the activity clipboard in the interaction record
8. Agent ends the interaction and he/she is now ready to place the next call.

In the default IC WebClient solution, the above scenario requires the agent to navigate to multiple screens as follows:

- Screen 1: Call list page
- Screen 2: Account Overview
- Screen 3: CRM Sales Order
- Screen 5: Interaction Record

Each of these screens is accessed in the standard IC WebClient via the navigation bar. After the navigational bar link is selected, it takes multiple clicks to execute a particular function. In order to obtain the total time it takes to execute each one of these functions, we need to take into consideration the mouse moving time to reach the appropriate navigational bar link plus the response time of each single click.

We concluded that the standard process can be optimized or enhanced by combining multiple functions in the same view. This view can be assigned to the IC business role as the default view via configuration.

In further sections we will demonstrate best practices to create a view like this, and provide example code.

### 1.4 Basic Setup

#### 1.4.1 CRM-ECC Connection

The example code and configuration provided in this cookbook is relevant to a CRM Stand-alone scenario using the standard CRM Order Management transactions and functions. However, this functionality can also be deployed in a CRM-ECC integrated scenario. If the latter is used, then you need to make sure that the CRM client is connected to an ECC client via middleware set up.

You can use the following Best Practice documents to configure the CRM-ECC connection:

- **C71: CRM Connectivity**
- **C73: CRM Misc. Connectivity**
- **B01: CRM Generation**
- **B09: CRM Customizing Replication**

These documents can be found at: http://help.sap.com/saap/sap_bp/CRM_XX/HTML/index.htm → Technical Information → Content Library
1.4.2 Customizing

The scenarios demonstrated in this cookbook are ‘Interaction Center – Sales’ with outbound call list functionality.

The following needs to be configured as a minimum in order to start the required enhancements demonstrated in this document:

- Master data: Customers, contact persons and products
- Interaction Center business role Sales Scenario with the following function profiles assigned:
  - Call list function profile
  - ERP Quote and Sales order function profile (only for CRM-ECC Connected scenarios)
- Call list configuration completed
- Configuration completed for CRM Sales Order Management (for stand-alone scenario), or ERP Quote and Sales order scenario (for CRM-ECC Connected Scenarios)
- CRM back-end transactions completed: Interaction Record and Call

1.5 Example Solution Realization

The code provided in this section serves only as an example that can be used by SAP customers in order to realize the logic to be followed to deploy their specific use cases. This code is not part of the SAP standard packages.

1.5.1 Preparation

1.5.1.1 Technical Requirements

Knowledge of SAP CRM IC WebClient Technology

1.5.1.2 UI Screens

- Account Overview screen
- Call list
- Interaction History
- CRM Sales orders

Below are some UI screens that can be used by customers to implement a similar tile layout page:

- Service Requests
- Service Orders
- Service Contracts
- ERP Sales Orders
- ERP Quotes
- Knowledge Articles
- Complaints
- Business Activities and Tasks
- Products
- IBase/iObjects
- Etc

1.5.1.3 UI Components
- Custom master component
- Standard window: ‘MainWindow’
- Inbound and outbound plugs
- Standard overview page: ClmCallListLeanOrderOVP
- Custom and standard component usages
- Standard views
- Standard context nodes: BTADMINH, BTADMINI, BTITEMS, BTODER and PARTNER

Customers can use other standard overview pages depending on the scenario they are implementing. Developers can make use of the F2 functionality in the Web UI page in order to find out what is the component name and page you need to include or use in the tile layout overview page. For example: if you were to determine the technical information of the view below:

You will need to place the cursor in any field and do an F2. The system will provide the following pop-up:
This pop-up will allow you to determine the Component name, View, Context Node, Attribute name and other necessary information.

You will need to determine what context nodes to use depending on your specific scenario. In order to determine this, you can make use of transaction GENIL_MODEL_BROWER, and select Component Set = ‘ONEORDER’ as follows:

Once inside the oneOrder model, you can expand the ‘Root Objects’ and determine the right object depending on the view you are working with:
1.5.2 Solution Mock-up

View 1: Major components in scope (high level review)

<table>
<thead>
<tr>
<th>Customer Name and Contact (Persistent Information)</th>
<th>Alert Area</th>
<th>Telephone Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call List</td>
<td>Call selected</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Confirmed Business Partner Overview</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales Order</td>
<td>Sales Order Items</td>
<td></td>
</tr>
</tbody>
</table>

1.5.3 High Level Technical Rationale

The code provided below is based on a particular scenario: ‘Outbound Sales in the Interaction Center’. The purpose of providing this code is to share the principles of combining multiple UI components with a tile layout in the interaction center following best practices. Every tile represents an application component: Call list, Business partner overview, Sales order and sales order items.

1.5.4 Step_by_Step Deployment of the Example Solution with Code Examples

1.5.4.1 Creation of an overview page with a tile layout

1. Create a custom master component using transaction BSP_WD_CMPWB

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>BSP_WD_CMPWB</th>
</tr>
</thead>
</table>

**UI Component Workbench**

- Component: [SAPCOOKBOOK]
- Test Application:

[Display] [Create] [Create Using Wizard]
2. Create a window with name “MainWindow” under master component ‘ZICCOOKBOOK’

![Structure of Component ZICCOOKBOOK - Standard View](image)

3. Add Model BTBP to Runtime repository

![Windows](image)

The BTBP model supports the BOL objects that are in scope to support the example we are using in this cookbook. If you were to use another component set (other than oneOrder), or a subset of the oneOrder component set, then you will need to use its corresponding model.

4. Define the window ZCOOKBOOK/MainWindow under the Runtime Repository / ‘ComponentInterface’, so the window is available within the value help:
5. Create context nodes: BTADMINH, BTADMINI, BTITEMS, BTODER and PARTNER with their respective base entities:

The context nodes displayed above support the use of the views that are in scope for the example used in this cookbook. However, if you were to use other views, you would need to bring their corresponding context nodes. For example: If you were to use the service order view, then you would need to use context note: BTHEADERTEXSET
6. Create custom controller: ZICCOOKBOOK/BP

The custom controller is responsible for binding all the views assigned to the same component. This is a persistent value used by all the views under the same component. If you were to implement another scenario, you will have to create your own custom controller. If you decide to implement the same scenario explained in this example, but using more views, then you will need to add their respective context nodes to this same custom controller.

7. Create BTORDER context node, and bind BTORDER context node to Component context BTORDER:

Other context nodes may be needed depending on the solution you are implementing.

8. Create BPHEADER context node, and bind BPHEADER context node to component context PARTNER:

Other context nodes may be needed depending on the solution you are implementing.
9. In order to allow for navigation to the detailed screens of the respective transactions, underneath the ‘MainWindow’, you need to create **outbound plugs** follows:

- TOBT
- TOACTIVITY

Other outbound plugs may be necessary depending on the specific navigation needed in your own implementation. For example: If you were to navigate from the order line items directly to the product view, then you will need to add the TOPRODUCT outboundPlug to the window.

10. Underneath the master component ZICCOOKBOOK, assign overview page called: ‘ClmCallList LeanOrderOVP’
In this specific example, we use the ‘ClmCallListLeanOrderOVP’ overview page as the tile layout page that contains all views that are in scope. You will need to add your specific overview page as a view to your specific component if you were implementing a different scenario.

11. Attach the new overview page ‘ClmCallListLeanOrderOVP’ to window ‘MainWindow’ as follows:
12. Register Overview Page: ‘ClmCallListLeanOrderOVP’ for Tile Layout as follows:

<table>
<thead>
<tr>
<th>SAP Menu</th>
<th>SAP Customizing Implementation Guide → Customer Relationship Management → UI Framework → UI Framework Definition → Register Overview Pages for Tile Layout</th>
</tr>
</thead>
</table>

![Display IMG]

- You need to assign the ‘CLMCALLLISTLEANORDEROVP.DO’ overview controller page key to the newly created BSP Application: ‘ZICCOOKBOOK’ as follows:
1.5.4.2Creation of component usages

This activity is performed in order to make sub_components (call lists, sales orders, and Business Partner details) available to the master component.

1. Create a call list component usage “ZCALLLIST” using component ‘CRMCMP_CLM’ and Interface View = ‘MainWindow’

Component usages need to be created according to the views we are using. The component usage facilitates the attachment of the individual views to the overview page. In this specific example, we are using CRMCMP_CLM because this is a call list component. In case you utilize other components, you need to use the corresponding component usage. This is found via the value help. For example, if you were to use component usage BTACTIVITY, then you will need to use component usage BT110H_ACT.

2. Create a sales order component usage “ZSLORD” using standard component ‘BT115H_SLSO’ and Interface View = ‘MainWindow’

3. Map inbound Plug ‘TOHEADER’ to the ‘MainWindow’ in order to make it available to the main component for the sales order header as follows:
4. Assign standard component usage 'BP_BPBT' as follows:

Final results:
The result screens above shows not only the component usage BP_BPBT that is added during this step, but also the all the components usages that are added at a later steps.
1.5.4.3 Add standard windows to master component 
ZICCOOKBOOK

1. Account Activities window:
BP_BPBT → AccountActivities
Inbound plug: – DEFAULT
Outbound plugs:-
- OutboundPlug BT Delegated to Window Outbound Plug TOACTIVITY
- OutboundPlug TOBT
- OutboundPlug TOCONTACT
- OutboundPlug TOEMPLOYEE

2. Account Sales Orders window:
Add interface view BP_BPBT/AccountSalesOrders
- OutboundPlug TOBT Delegated to Window Outbound Plug TOBT
3. **Account Interaction History window:**

Add interface view `AccountInterHist`

- OutboundPlug BT Delegated to Window Outbound Plug `TOBT`
4. **Account Header window:**

Add component usage ‘AccountHeader’
Add interface view BP_HEAD/AccountHeader

Structure of component ZICOOKBOOK - Standard view

- Component Structure Browser
- BOL Model Browser
- Runtime Repository Editor

Component
  - Models
  - Windows
    - Window ZICOOKBOOK/MainWindow (default)
      - InboundPlugs
      - OutboundPlugs
        - OutboundPlug TOACTIVITY
        - OutboundPlug TONIT
    - View ZICOOKBOOK/CmCl钤ListLeaOrderDT (default)
  - ViewSets
  - NavigationLinks
  - ComponentInterface
  - ComponentUsages
    - ComponentUsage ZCALLLIST
    - ComponentUsage ZSLSORD
    - ComponentUsage BP_BPRT
    - ComponentUsage AccountHeader
      - Used Component BP_HEA
        - InterfaceView BP_HEAD/AccountHeader
          - InboundPlugs
          - OutboundPlugs
Final Results:

Windows are added to the main component depending on the views that you are adding to the tile layout main component. This may be different in your specific example or in your use case.
1.5.4.4  Create Component usage for the sales order items

1. Add component usage "CUItems" to 'ZICCOOKBOOK/MainWindow'

2. Add component BT115IT_SLSO with Interface view: 'MainWindow'. In addition, add the following inboundPlugs and OutboundPlugs:

   - InboundPlug FORHEADERLIST
   - OutboundPlug NAVIGATE_DISPLAY
   - OutboundPlug PROD_CONFIG
   - OutboundPlug TOHEADEROVFROMITEMEL
   - OutboundPlug TOITEMOV
   - OutboundPlug TOITEMOVFROMITEMEL
   - OutboundPlug TOITEMOVFROMNEWITEMEL
Consultant's Cookbook Interaction Center CRM 7.0 EHP1 and above - Maximizing User Experience

Component Structure Browser
BOL Model Browser
Runtime Repository Editor

Component
- Models
- Windows
- ViewSets
- NavigationalLinks
- ComponentInterface
- ComponentUsages
  - ComponentUsage ZCALLLIST
  - ComponentUsage ZLSORD
  - ComponentUsage BP_BPEIT
  - ComponentUsage AccountHeader
  - ComponentUsage CUIsense
    - Used Component BILLISIT_SLSO
    - InterfaceView MainWindow
- InboundPlugs
  - InboundPlug FORHEADERLIST
- OutboundPlugs
  - OutboundPlug NAVIGATE_DISPLAY
  - OutboundPlug PROD_CONFIG
  - OutboundPlug TOHEADEROVRITEMEL
  - OutboundPlug TOITEMOV
  - OutboundPlug TOITEMOVERITEMEL
  - OutboundPlug TOITEMOVERNEWITEMEL
1.5.4.5 Make all component usages available to the overview page

Overview page: ‘ZICCOOKBOOK/ClmCallListLeanOrderOVP’

1. Add component usages created in step 4 to the overview page “ClmCallListLeanOrderOVP”
2. Create UI configuration in overview page “ClmCallListLeanOrderOVP”
   - Create following tiles
     - Call list
     - Account details
     - Sales order
     - Sales order item

   - Add component usage “ZCALLLIST” to tile “Call list”
   - Add Component usage “AccountHeader.BP_HEADER/AccountHeader” to tile “Account Details”
   - Add Component usage “BP_BPBTC.AccountActivities” to tile “Account Details”
   - Add Component usage “BP_BPBTC.BP_BPBTC/AccountSalesOrders” to tile “Account Details”
   - Add Component usage “BP_BPBTC.AccountInterHist” to tile “Account Details”
   - Add Component usage “ZSLSORD.MainWindow” to tile “Sales Order”
   - Add Component usage “CUItems.MainWindow” to “Sales Order Items”

The screen above shows a tile layout structure that is not part of the standard delivery. This tile layout was configured as follows:

<table>
<thead>
<tr>
<th></th>
<th>Tile 1</th>
<th>Tile 2</th>
<th>Tile 3</th>
<th>Tile 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can also use the standard tile layout, which can be utilized without further activities required. This layout comes in the standard as follows:

<table>
<thead>
<tr>
<th></th>
<th>Tile 1</th>
<th>Tile 2</th>
<th>Tile 3</th>
<th>Tile 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.5.4.6  Redefine methods

You will need to redefine the method DO_CONFIG_DETERMINATION in the overview page in order to have the system differentiating between customers and contacts. The code below shows how this is implemented for customers only:

a. Redefine method DO_CONFIG_DETERMINATION in overview page “ClmCallListLeanOrderOVP”

The purpose of redefining this method is to bind the confirmed business partner to the new custom controller.

Depending on the scenario you are implementing, you may need to redefine a different method. For example: in our specific scenario we could also use the ‘DO_PREPARE_OUTPUT’ method as well.

Add the following example code to method DO_CONFIG_DETERMINATION

```abap
DATA declarations
    lr_bp_entity type ref to cl_crm_bol_entity,
    lr_order   type ref to cl_crm_bol_entity,
    lr_btitem  type ref to cl_crm_bol_entity,
    lv_entity_rel type ref to cl_crm_bol_entity,
    lr_collection type ref to if_bol_bo_col,
    lr_entity_admin type ref to if_bol_entity_col,
    lr_iterator type ref to if_bol_entity_col_iterator,
    ls_viewid type bsp_dlc_ovw_stat_view_attach,
    entity     type ref to cl_crm_bol_entity,
    lv_config  type string,
    lr_cust_controller type ref to ZL_ZICCOOKB_BP_IMPL.

DATA: lr_bdc TYPE REF TO if.crm.ui.data_context,
      lr_bdc_controller TYPE REF TO cl_bsp_wd_controller,
      lr_customer TYPE REF TO cl_crm_bol_entity,
      lr_contact TYPE REF TO cl_crm_bol_entity,
      ls_startsession TYPE crmt_ic_re_startsession,
      lv_startsession TYPE string,
      lv_event_container TYPE crmt_ic_re_event_container.
DATA lv_timestamp TYPE string.

DATA lr_cucobupa TYPE REF TO cl.crmcmp_b_cucobupa_impl.
```
try.

*get custom controller

    lr_cust_controller ?= get_custom_controller( 'ZICCOOKBOOK/BP').

    if lr_cust_controller is bound,
        if lr_cust_controller->typed_context is bound,
            if lr_cust_controller->typed_context->bpheader is bound,
                if lr_cust_controller->typed_context->bpheader->collection_wrapper is_bound,
                    endif.
                endif.
            endif.
        endif.
    endif.

    if lr_bp_entity is not bound,
        lr_bdc = cl_crm_ui_data_context_srv=>get_instance( ).
        lr_bdc_controller ?= lr_bdc.
        lr_cucobupa ?= lr_bdc_controller->get_custom_controller( if_iccmp_global_controller_con=>cucobp ).

    * if bp has been confirmed already then add to the custom controller
        IF lr_cucobupa->is_bp_confirmed( ) EQ abap_true.

    * get the current customer
        lr_customer ?= lr_bdc->get_entity( name = 'CURRENTCUSTOMER' ).

        IF lr_customer IS Bound.
            CREATE OBJECT lr_collection TYPE cl_crm_bol_bo_col.
            lr_collection->add( lr_customer ).
            lr_cust_controller->typed_context->bpheader->SET_COLLECTION( lr_collection ).
            ENDIF.
        endif.
    endif.
endif.
catch cx_root.
endtry.

CALL METHOD super->do_config_determination
    EXPORTING
        iv_first_time = iv_first_time.

b. Create two Event Handlers one for new order and another for to save the order as follows:

- **EH_ONNEW_SLS_ORD**
- **EH_ONSAVE_SLS_ORD**
**EH_ONNEW_SLS_ORD** allows the users to create and save sales orders from the overview page.

**EH_ONSAVE_SLS_ORD** will be called when the user selects 'New' button

C. Add the following sample code to method **EH_ONNEW_SLS_ORD**

```abap
DATA:
  lt_params  TYPE crmt_name_value_pair_tab,
  ls_params  TYPE crmt_name_value_pair,
  lr_current  TYPE REF TO if_bol_bo_property_access,
  lr_core    TYPE REF TO cl_crm_bol_core,
  lr_factory TYPE REF TO cl_crm_bol_entity_factory,
  lr_order   TYPE REF TO cl_crm_bol_entity,
  lr_cuco    TYPE REF TO ZL_ZICOOKB_BP_IMPL,
  lr_entity  TYPE REF TO cl_crm_bol_entity.

DATA: lr_access type ref to IF_CRM_UIU_bt_CHANNEL_ASPECTS.

FIELD-SYMBOLS: 
  <ls_data>  TYPE ANY.
```
* set process type
  ls_params-name = 'PROCESS_TYPE'.
  ls_params-value = 'ZTSA'. "Hard coded this needs to be adjusted by
the customer"
  APPEND ls_params TO lt_params.

* Create Root Entity
  lr_core = cl_crm_bol_core=>get_instance( ).
  lr_factory = lr_core-
  get_entity_factory( 'BTOrder' ). "#EC NOTEXT
  lr_order = lr_factory->create( lt_params ).
  CHECK lr_order IS BOUND.

  lr_access = cl_crm_uiu_bt_channel_asp_fac=>get_instance( ).
  lr_access-
  if_crm_uiu_channel_aspects-after_create( iv_entity = lr_order ).

  * get custom controller
    lr_cuco ?= get_custom_controller( 'ZICOOKBOOK/BP').
    if lr_cuco is bound.
  * Get rid of the obsolete entity, so it does not create problems late
    r, eg breadcrumbs
    lr_cuco->typed_context->btorder->collection_wrapper->clear( ).
  * Add the new BTOrder-Entity to collection
    lr_cuco->typed_context->btorder->collection_wrapper->add( 
      iv_entity = lr_order
      iv_set_focus = abap_true ).

    lr_order->switch_to_change_mode( ).

  endif.

endmethod.
d. Add the following sample code to method **EH_ONSAVE_SLS_ORD**:

```abap
method EH_ONSAVE_SLS_ORD.
  DATA: lr_cn TYPE REF TO cl_bsp_wd_context_node,
        lr_ent TYPE REF TO cl_crm_bol_entity,
        lr_cuco type ref to ZL_ZICCOOK_BP_IMPL.
  try.
    * get custom controller
      lr_cuco := get_custom_controller( 'ZICCOOKBOOK/BP' ).
      if lr_cuco is bound.
        lr_cn = lr_cuco->typed_context->btorder.
    * execute sync save
      cl_crm_uiu_bt_tools=>save( lr_cn ).
      lr_ent := lr_cn->collection_wrapper->get_current().
    * prefetch data after save
      * remove and add entity in order to refresh all context nodes (selections, etc.)
      * and lock registration is done
      lr_cn->collection_wrapper->clear().
      lr_cn->collection_wrapper->add( iv_entity = lr_ent,
                                    iv_set_focus = abap_true ).
    endif.
    catch cx_root.
  endtry.
endmethod.
```

e. In order to make these buttons (New and Save) visible in the UI, you will need to redefine the following methods overview page “ClmCallListLeanOrderOVP” as follows:
- IF_BSP_WD_TOOLBAR_CALLBACK~GET_BUTTONS
- IF_BSP_WD_TOOLBAR_CALLBACK~GET_NUMBER_OF_VISIBLE_BUTTONS

Add the following sample code to method:

```abap
IF_BSP_WD_TOOLBAR_CALLBACK~GET_BUTTONS
```
method IF_BSP_WD_TOOLBAR_CALLBACK~GET_BUTTONS.
DATA:
  ls_button TYPE crmt_thtmlb_button_ext.

* New Sales Order
  ls_button-text = 'Create Order'. 
  ls_button-on_click = 'NEW_SLS_ORD'.
  ls_button-page_id = me->component_id.
  ls_button-enabled = abap_true.
  APPEND ls_button TO rt_buttons.
  CLEAR ls_button.

* Save Order
  ls_button-text = 'Save Order'. 
  ls_button-on_click = 'SAVE_SLS_ORD'.
  ls_button-page_id = me->component_id.
  ls_button-enabled = abap_true.
  APPEND ls_button TO rt_buttons.
  CLEAR ls_button.

endmethod.

➢ Add the following sample code to method
  IF_BSP_WD_TOOLBAR_CALLBACK~GET_NUMBER_OF_VISIBLE_BUTTONS

method IF_BSP_WD_TOOLBAR_CALLBACK~GET_NUMBER_OF_VISIBLE_BUTTONS.
  RV_RESULT = 2.
endmethod.

f. Create the following outbound plugs in component MainWindow
   ▪ OP_TOACTIVITY
   ▪ OP_TOBT

➢ Add the following code to method OP_TOACTIVITY
  method OP_TOACTIVITY.
    me->fire_outbound_plug( iv_outbound_plug = 'TOACTIVITIES'
    iv_data_collection = iv_data_collection ).
endmethod.

➢ Add the following code to method OP_TOBT
  method OP_TOBT.
    me->fire_outbound_plug( iv_outbound_plug = 'TOBT'
    iv_data_collection = iv_data_collection )
  endmethod.
1.5.4.7 Configure Navigation Bar Profile

1. Configure the navigation bar profile as follows:

<table>
<thead>
<tr>
<th>SAP Menu</th>
<th>Transaction Code</th>
</tr>
</thead>
</table>

The navigation bar profile assigned to our ZCOOKBOOK_IC business role is ‘ZCOOKBOOK’. See assignment of navigation bar to business role below using transaction: CRMC_UI_PROFILE

2. In transaction: CRMC_UI_NBLINKS, create a new navigation bar profile as a copy of the standard ‘IC_AGENT’ navigation bar:

3. Create a new target ID view with it's inbound and outbound plugs

<table>
<thead>
<tr>
<th>SAP Menu</th>
<th>Transaction Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Customizing Implementation Guide → Customer Relationship Management → UI Framework → Technical Role Definition → Define Work Area Component Repository</td>
<td></td>
</tr>
</tbody>
</table>
a. Create a new component definition for our window 'ZICCOOKBOOK/MainWindow as follows:

![Component Definition](image)

b. Create a new inbound plug definition for our custom component 'ZICCOOKBOOK'. There’s no need to create outbound plugs:

![Inbound Plug Definition](image)

4. Create a new logical link and workcenter as follows:

![Logical Links](image)

a. Select ‘Define Logical Links’, and create a new entry:

b. Assign the new Target ID ‘ZLEANORDER’ created in step 3 above:
c. Create a new Work Center and assign the new logical link ‘ZICBOOK’:

![Change View "Define Logical Links": Details](image)

- Create a new Work Center and assign the new logical link ‘ZICBOOK’.
- Assign the newly created work center ‘ZCOOKBOOK’ to our navigation bar: ‘ZCOOKBOOK’ as follows:

![Change View "Define Work Centers": Details](image)

- Assign the newly created work center ‘ZCOOKBOOK’ to our navigation bar.
- Make sure that the newly created navigation bar profile is assigned to the relevant business role.

![Change View "Assign Work Centers To Navigation Bar Profile": Overview](image)

- Make sure that the newly created navigation bar profile is assigned to the relevant business role.
1.5.5 **Solution Results**

Final view in the interaction center:
1.6  Improving call efficiency by using UI Shortcuts and Intent Driven Interactions:

The SAP/CRM Interaction Center WebClient provides multiple ways to increase efficiencies without having to enhance the application. Two standard functions that can be used to achieve this goal are:

- Setting UI Shortcuts at user level, and
- Configuring intent driven interactions

Find below more details on these functions.

1.6.1  UI Shortcuts

IC agents can set up shortcuts directly in the interaction center.

**Pre-requisite:** IC WebClient business role in the customer name space needs to exist, and the user needs to be assigned to this business role either via the organizational model or using parameter CRM_UI_PROFILE in the user record (Transaction: SU01.)

Please follow the steps below to customize your shortcuts:

1. Log in to the business role:

2. Click on the ‘Personalize’ link:
3. Select ‘Personalize Shortcuts’:

4. Select the shortcuts you want to personalize:
Keep in mind that these changes will only be reflected in the specific user profile since this is performed via personalization and not via configuration.

### 1.6.2 Intent Driven Interactions

The Intent Driven Interaction context provides the ability to create alerts in the interaction center.

**Prerequisites:**

- You have completed the Customizing for Customer Relationship Management under Interaction Center WebClient → Additional Functions → Intent-Driven Interactions → Service Manager → Configure Alerts
- You have completed the Customizing for Customer Relationship Management → Interaction Center WebClient → Additional Functions → Intent-Driven Interaction → Define Intent-Driven Interaction Profiles
- If the alerts you are using also refer to information in SAP NetWeaver BI, you must have also set up and configured this component. For more information about configuring alerts, see SAP Solution Manager
- If you want to use user-triggered alerts, you have selected this alert type in Customizing for Customer Relationship Management → Interaction Center WebClient → Basic Functions → Communication Channels → Define Communication Processing Profiles
Alerts are freely definable, and they are triggered and terminated via business rules. The tool that is used to create the rules to trigger alerts is the Rule Modeler.

Alerts are triggered by events, and once the alert is displayed, the interaction center provides the ability to navigate to another page by clicking directly on the alert.

Please see below an example of an alert that is triggered based on the category selected in a service request:

**Use Case:**

- Service requests are created in the interaction center to capture customer issues.
- Service requests are categorized using a hierarchical categorization schema.
- Knowledge articles are created to represent solutions to common problems.
- In order to promote efficiencies in the interaction center, knowledge articles are assigned directly to specific categories, so when the agent selects that category in the service request and during the course of the interaction with the customer, the relevant knowledge article gets proposed via an alert.
- The agent can click on the alert, and navigate directly to the relevant knowledge article for review.
Pre-requisites:

1. Categorization schema needs to exist and assigned to the appropriate service request type

![Categorization Schema](image)

2. Knowledge article(s) needs to exist

![Knowledge Article](image)

**Note:** The categorization schema assigned to knowledge articles can be different than the one assigned to the service requests. The scenario we are demonstrating with this example does not require a categorization schema assigned to the knowledge article. In order for this scenario to work, a categorization schema needs to be assigned to the service request only. The relevant knowledge article has to be assigned to a category.

3. Knowledge article(s) need to be assigned to the appropriate category in the categorization schema
Procedure:

1. Configure alert
   a. Access the activity using the following navigation option:

   **SAP CRM WebClient UI** | **Business Role: IC Manager**
   **Process Modeling → Alert(s)**

   b. SAP provides a standard ‘AUTOSUGGEST” alert that can be used by customers. If you need to change the content of this alert, then make a copy of it, or create your own using the customer name space:

2. Configure rule using the rule modeler
   a. Access the activity using the following navigation option:

   **SAP CRM WebClient UI** | **Business Role: IC Manager**
   **Process Modeling → Create: Rule Policy**

   b. Create a rule policy as follows:
c. Assign the appropriate IC WebClient business role(s) and the appropriate events as follows:

![New Rule Policy -- Webpage Dialog](image1)

---

d. Create 2 rules as follows:

Rule 1:

![Rule Policy: ZALERTKAEXIST (Intent Driven interaction (IC WebClient))]
Rule 2:

Final Results:
1. Access the activity using the following navigation option:
   - SAP CRM WebClient UI
   - Business Role: Customer’s IC_Agent business role
     - In this specific example is: BP_ICA_SRV

2. Select your customer and confirm the account. Then create a service request and select the following categorization: Request (first category) → Online Shop (2nd category). As you can see below, the alert shows up showing the number of knowledge articles that are assigned to this category:
3. If the user clicks on the alert, the application will navigate to the relevant knowledge article:
1.7 Appendix

1.7.1 Helpful Links

Customer Relationship Manager  →  SAP CRM
SAP CRM – Installation and Upgrade Information:  service.sap.com/crm-inst
→  Content Library
Solution Manager Diagnostics (SMD, see  http://service.sap.com/diagnostics).
SAP Netweaver AccAD details:  http://scn.sap.com/community/accad-for-netweaver

1.7.2 CRM Performance Analysis and Best Practices

1.7.2.1 Preparation and Purpose

This chapter provides an overview and summarizes the best practices for performance analysis and architecture in order to make sure that the CRM solution runs at its best performance levels and with the lowest maintenance. This chapter is a collection of findings and best practices collected over time, and it is not meant to be complete nor part of the delivery.

1.7.2.2 Performance Analysis

In this sub-chapter, you will find some information on how to do a successful analysis of any performance issue reported by the end users. First of all, the end user identifies the performance problem by a long response time of the system, often combined with an active progress indicator (e.g. spinning wheel). The root cause of the issue can be at many spots within the solution, not only the CRM system:
The list of components contains among others: Client PC itself, including the browser, network, network components like switches, routers, firewalls, proxies, the CRM Server(s), DB, connected SAP systems like ERP, 3rd Party systems.

The tool of choice for the performance analysis would be the Solution Manager Diagnostics (SMD, see http://service.sap.com/diagnostics). It allows an end-to-end analysis starting on the frontend. In this document we show how to do a performance analysis if the SMD is not available. Here we will focus on the CRM server incl. DB, the network components and the client PC:

In order to find a first start for the analysis let the user execute the slow transaction in the CRM WebClient. Then display the respective statistical record with transaction STAD:

Example:

<table>
<thead>
<tr>
<th>Started</th>
<th>Server</th>
<th>Transaction</th>
<th>User</th>
<th>Response Time (ms)</th>
<th>Time in CPU (ms)</th>
<th>Time in DB Request (ms)</th>
<th>Wait Time (ms)</th>
<th>GPU Time (ms)</th>
<th>DB Request Time (ms)</th>
<th>VMC Elapsed Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:19:29</td>
<td>acbe908_001_761</td>
<td>335000</td>
<td>8</td>
<td>22.868</td>
<td>20.606</td>
<td>8</td>
<td>8.905</td>
<td>11.811</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
The performance problem can have multiple root causes. The best approach is to focus first on the longest running contributors to the overall response time of the critical step. The main contributors can be split into:

- Client PC
- Network
- ABAP Processing
- DB
- RFC

Follow this simple approach:

- Is the total response as seen in STAD significantly lower than the response times that the user reports (e.g. users says it takes 60 seconds but STAD shows 22 seconds total response time)?
  - Check the Client PC and the network first
- Is the CPU time the longest contributor
  - Do an ABAP performance analysis
- Is the DB time the longest contributor
  - Do an SQL trace analysis
- Is the RFC time the longest contributor
  - Check if it is CRM internal (like VMC) or outbound to other SAP or 3rd party system. Continue the performance analysis there.

**Network and client PC**

In this sub-chapter, we will discuss the main transactions and tools for network performance analysis. We will give recommendations/best practice guidance for the most important settings.

**Client PC and browser**

The largest share of the total response time lost on the client PC is usually spent on rendering. Within the browser the business pages based on HTML need to be displayed and with varying complexity this can take up to 500 ms or longer. In order to determine if rendering is taking too long, start in parallel to the CRM WebClient the Windows Task-manager and check the CPU usage while the page is rendered. If one of the CPU cores is up to 100% for longer time (> 250-500 ms) the reasons can be the following:

- The CPU on the client PC is too weak. We recommend upgrading to at least a Dual-core CPU as it is our experience that on a Windows PC other tasks are often running in parallel (virus scanning, email-software, music playing.)
- Outdated Browser version: As IE 6 is not supported by MS anymore we strongly recommend upgrading to newer versions. Nevertheless, IE7 is showing 50% slower rendering times than the successor versions IE8-10. The performance has been improved by MS with newer browser versions
- Virus scanner sometimes provides script scanning functionality on Web Pages. We have seen in some cases rendering times going from 300 ms without Script Scanning to up to 6 seconds with active scanning. Check with your Virus scanner provider if newer versions show better performance of if the CRM server can be excluded from scanning.
Compression

HTTPWatch or MS Fiddler can be used to do a trace of the HTTP traffic. In general a round trip in CRM WebClient creates traffic of about 5-20 Kbyte depending on the screen content. A main reason for bad network performance can be a lack of compression which lets the volume increase by up to a factor of 20. HTTPWatch and Fiddler can show such a problem.

In an environment where there is no central control over the connected client PCs at the dealer sites we have to limit the settings' recommendations to a minimum.

In the client PC it is mandatory that HTTP1.1 is enabled otherwise compression is not supported.

Make sure that the cache of the browser is active but not too large. Otherwise the local search for stored mime objects can take longer than the download.

Follow Microsoft's recommendation of not having more than 250 MB of local cache, 100 MB is more than sufficient.

On the CRM server side there are several points to look at:

Profile Parameter

In order to improve compression please add the following parameter to the profile of the instance (/usr/sap/xxx/SYS/profile/xxx_DVEBMGS00_xxxxx):

```
ict/exclude_compression =
*.zip,*.rar,*.arj,*.z,*.tar,*.lz,*,.cab,*,.hqx,*,.ace,*,.jar,*,.ear,*,.war,*.gzip,*,.uue,
```

The default setting which is active when the parameter is not set explicitly includes *.css and *.pdf (and in former version of NW also .js). Only add PDF if mainly pre-compressed PDFs are delivered e.g. via content management system. In this case additional compression would even increase the data volume.

ICF and BSP compression

ICM (Internet Connection Manager, the NW-Webserver) automatically compresses all content transferred in stateful mode (HTTP1.1). You may recognize inconsistent compression behavior in HTTPWatch, as a workaround you can try to activate the compression explicitly on the affected URI-nodes in ICF (Internet Connection Framework). Go to transaction SICF and open the affected node. You can explicitly activate the compression there:
This should only be a workaround until the root cause for the missing compression is found:

On the BSP side there is often a misunderstanding about which switches are responsible for the compression. Here it also valid that changes to a BSP, especially SAP delivered ones, should only be done by recommendation of SAP Support. The functionality to allow compressed transfer in the BSP is defined by the top level flag “stateful”. Only stateful BSPs can be transferred by HTTP1.1 which allows GZIP compression.

If one of the pages underneath this BSP shows no compression check the details of this page. Here it is important that the option “stateful” is not changed to “stateless from Now on”. “Unchanged” makes sure that HTTP1.1 is also used on this BSP-page.
The flag “compression” in the block “Transfer Options” is not making any difference as it is just still there for compatibility reasons with CRM 5.0 delivered PC-UI pages for portal integration.

**SMICM => Internet connection manager**

Each service is defined in ICM like this:

```
icm/server_port_0   = PROT=HTTP, PORT=50000, PROCTIMEOUT=600, TIMEOUT=90
```

Browsers (especially IE) have a fixed TCP keep_alive_timeout of 60 seconds. It therefore does not help to extend the value much more than that. A good recommendation is to have a value of 90 seconds for HTTP/HTTPS services. Otherwise the server would keep too many sockets open, and requests will never be sent to the client. This can even lead to a shortage of free TCP sockets on the OS level (Linux and MS Windows have usually only 2000 sockets per default). If TIMEOUT is not set explicitly on a service the global parameter ICM/KEEP_ALIVE_TIMEOUT is taken. It is important not to set a TIMEOUT < 60 secs otherwise unnecessary re-connects would take place. The keep_alive_timeout parameter is NEVER responsible for a TIMEOUT error message in the CRM WebClient.

It is mandatory to define a PROCTIMEOUT parameter in the service definition. Otherwise any TIMEOUT or global keep_alive_timeout setting will be ignored and a value of 30 seconds will be taken.
PROCTIMEOUT defines how long the Web-Interaction waits for a result from a DIALOG work process within the CRM Instance. If e.g. the user fires a query that takes 30 minutes to fulfill and the PROCTIMEOUT is 600 seconds (10 minutes) the user would get a TIMEOUT error message. The question is how long you expect your users to wait in front of the browser before giving up.

The parameter rdisp/plugin_auto_logout defines the time of maximum inactivity until a user session gets deleted. This depends on the business requirements; a value of 1-4 hours has proven to be useful.

In order to allow logging set the parameter icm/HTTP/logging_0 = PREFIX=/, LOGFORMAT=%t %a %u - %r %s %b %L, LOGFILE=http_log-%d, MAXSIZEKB=25000, FILEWRAP=on

### SAP Web Dispatcher

In case you are using an SAP WebDispatcher make sure that all timeout parameters are defined in sync with the application servers. The CRM Web Dispatcher can be used as a Web Cache for better performance.

See below for details:


### NetWeaver AccAD

Accelerated Application Delivery for SAP NetWeaver (AccAD) is a standalone product which is positioned as a complimentary offering to SAP NetWeaver. It can be leveraged as an enabler for SAP’s global central system strategy, e.g. when setting up one central portal which is the access point for all end users worldwide. Quite commonly, due to bandwidth and latency restrictions, users in remote offices encounter performance issues like lengthy response times for logging in or navigating within an SAP Solution. Accelerated Application Delivery is a software appliance that can overcome those issues and thus deliver applications at near-LAN speed to global user groups. Accelerated Application Delivery can provide benefits for all web-based SAP applications. It’s application aware compression and caching mechanisms can be more efficient than generic WAN acceleration and application delivery technologies. It focuses on accelerating global access to SAP applications and thus it is not intended to be a full-fledged holistic application delivery tool for all kinds of traffic.

See [http://scn.sap.com/community/accad-netweaver](http://scn.sap.com/community/accad-netweaver) for more details. Start there with the document “Get started with Accelerated Application Delivery for SAP NetWeaver”.

### ABAP Performance Analysis

In case the major part of the total response time is on the ABAP part of the application the analysis has to start on the CRM application server(s). First of all, it is good to check if any specific issues on the server might have caused the performance problems. This is done in less than 15 minutes and can avoid even more lengthy analysis by tracing.

**ST02, ST06, ST13**

In case the user reports fluctuating response times for the same transaction and long CPU times are found in STAD, check if the issue is caused by load & generation by double-clicking the respective row in STAD.
An SAP Application Server should provide all typically used transactions ready for use, and no compilation should be required. If the system was restarted recently or an upgrade / Note implementation took place, compilation should occur. Check transaction ST02 if there are any swaps, mainly in the area of program buffer.

If the response time is high but it is not obvious where the response time was lost (a “gap” in the figures), check in ST02 via details the semaphore screen. Hit the refresh button once. Sometimes, hardware issues on network cards can show up here making the processing time look long. In fact the system just waited for a hardware event to finish. Similar semaphores may occur if there are bottlenecks on enqueue processes. There is no rule of thumb but too many semaphores with times > 100 ms should not occur in the system. Otherwise check with the semaphore number for SAP notes.

Another reason for fluctuations may be bottlenecks in resources caused by parallel high load e.g. because of a mass data change. Go to transaction ST03, choose the date for when the user reported the issue and go to “Time Profile”. Change the view for the different transaction types (HTTP, DIA, RFC, …) and find out if any high load was executed in parallel.

By double-clicking on the suspect row you can drill down into the details all the way to the single statistical record. Be aware that with the standard settings statistical data is available 24h per record, and after that the data are aggregated by user. After a week and a month respectively, this gets aggregated more and more. Therefore, an early reaction to a performance issue is important.

In case users complain on standard navigation and transactions, the system may have a bottleneck in CPU resources. Check transaction ST06 (ST06n or OS07n in versions < NW 7.30) for details. Especially in virtualized environment (incl. LPAR) the total available CPU load does not say too much about a possible CPU bottleneck. When the entitlement is lower
than the vCPUs often the overhead on the Hypervisor causes delays that are not reflected in CPU consumption. Especially for CRM instances running online users the recommendation is to always use at least entitlement = vCPUs, better dedicated CPUs. A CPU consumption history can be found in ST06, too.

If you have doubts about the CPUs’ performance, take a look at SAP Note 1501701: Single Computing Unit Performance and Sizing.

Single Thread performance of older hardware is often insufficient for CRM Online users. SAP Quicksizer recommends AAA rated CPUs for CRM Online usage. This means that the single thread performance of a CPU has to be significantly higher compared to the requirements of e.g. an ERP system. The total number of SAPs is not affected by this. SAPs is a figure for throughput, not for performance.

Example measurement:

Usually in transaction ST13 the PERF_TOOL => ABAPMeter can be called. This requires installation of ST-API 01N. The tool delivers values that help to estimate the hardware performance of a server.

The first value in the result of this tool is calculated like this:

```
Runtime: 137,984 microseconds
```

Our experience from other customer installations shows the following:

A server that can execute the same code in half of the time shows an average improvement in ABAP processing times of 50%. This is independent from the benchmark value (which is 137 ms in the screenshot). Means: a server that can execute the same code in 60 ms is twice as fast in ABAP processing as a server that executes the same code in 120 ms.

Servers with CPUs with modern architecture can execute the same code already below 20 ms (e.g. X5670 @ 2.93GHz). CPUs like the E5-2690 have already about 2600 SAPs per CORE! ([http://www.sap.com/solutions/benchmark/sd2tier.epx](http://www.sap.com/solutions/benchmark/sd2tier.epx)). These are just examples and not a recommendation for a specific hardware or hardware vendor.

If you upgrade your app servers to more modern CPUs the users often immediately experience greatly improved performance. This is valid where the CPU ABAP time holds the larger share in total response time compared to DB time.

For testing we recommend that you add one app server on faster hardware to your productive landscape and add a few test users to it via logon group control.

In case you determine that the long ABAP processing time is neither caused by

- Application buffer swapping
- Parallel mass execution
- CPU bottleneck
- Slow CPUs
you have to take the next step and do an ABAP Trace.

**Transactions SE30/SICF/SAT**

Transaction SE30 is the “old” tool for ABAP performance analysis. It uses a so called “variant” which defines what is recorded in the trace. The DEFAULT setting is often completely sufficient. Traces are saved on disk level and can be retrieved with “Other File” option on the bottom of the screen.

After starting the transaction you see on the top left either a green or a red light. In case you see a red light you are using a UNIX operating system that does not provide a high precision time measurement. In that case change via the menu in SE30 to low precision and restart the application. SE30 only runs on the current instance, not cross-instance.

In order to start SE30 for WebClient based transactions you have two options (there are more, but others are often too difficult to pursue): either activate the trace in SICF on the CRM WebClient node (CRM_UI_FRAME in standard) or start it on the WebClient itself.

**SICF:**
When you activate “single entries” you will have an own trace file in SE30 for every round-trip.

**WebClient:**
Start the performance analysis box with CTRL+SHIFT+F7. Before you execute the long running step, activate the runtime trace:

It is not possible to choose a different variant than “DEFAULT”. It is always a composite entries recording. There is no trace stop option.

If you have the Solution Manager Plugin tools (ST-A/PI) installed you also have access to ST12. This transaction allows the start of the ABAP trace on the backend for a specific user. It can activate an SQL performance trace in parallel. The result for ST12 is similar to the SE30 result. It provides additional call stack information.

The successor of SE30 is the Single Activity Trace SAT (fully available since NW 7.02): It provides a new user interface with a similar look and feel like the ABAP New Debugger. Traces are no longer stored on disk level but in the database and have an automatic expiration time which can be extended on demand.

In all cases you would check the trace for the largest Net contributors (quick wins) first:

Identifying the longest running calls allows a limitation of the root cause to a specific component on where to open a customer message (e.g. CRM-RFW, CRM-IC…). In case you encounter Z-Code as the called or calling function module, contact your consulting support.
DB Performance Analysis

If you think that a slow transaction step is mainly caused by one or more long running SQL statements do an SQL trace of the process step.

For this start ST05 (attention: only valid for the current instance you are logged on to). Activate the SQL trace via the “activate with filter” in case you have to trace another user and not your own:

Let the user execute the long running statement. If possible this should only cause one roundtrip (=one click in WebClient).

Deactivate the trace and press “Display Trace”.

Summarize the trace by SQL statement:

By double-clicking a statement you see the parameters of the Where clause:
By pressing the EXPLAIN button you can then see the execution plan of Oracle and which index/indices have been used (in the example no index was chosen but a full table scan was done. This requires an urgent investigation):

The button next to the ‘Explain’ button shows the ABAP code where the statement is created. This should help you find the correct component, and where to open a customer message if it is required. It also help you identify the responsible development team in the case of a Z-code.

**RFC Performance analysis**

Details on an RFC connection can be found when you double-click an STAD record and either click on RFC on top or scroll down through the according block.

The total number of RFC calls related to this call are shown. For process steps in WebClient, RFC records can be found mainly ‘as client’, and seldom ‘as server’.

By clicking on the green field, details of the connections are shown. Please note that there are asynchronous calls which don’t contribute much to the overall runtime and synchronous calls which are counted completely.
Attention! On the details, only the 5 longest running RFC connections are shown, and not all! In case the performance issue is caused by a call to an external system, execute the performance analysis there. If it is an SAP Netweaver ABAP system, you can start by checking the STAD records. Username and system can be found in the details as shown above. If it is e.g. a CRM internal call (e.g. to VMC) continue the analysis by changing the STAD selection criteria to RFC and the username to the RFC user as shown above.

1.7.2.3 Middleware

In a standard system configuration, middleware queue traffic is managed by queue schedulers. There is a unique scheduler for outbound queues and a unique scheduler for inbound queues. Each scheduler has the option of using a specific application server group or all available application servers for queue processing. Using an application server group allows each scheduler to load balance queue processing across several pre-designated application servers. The load balancing is done at queue scheduler level and not based on individual queues. This is sufficient for most implementations.

These logon groups for the scheduler can be defined in RZ12:
Example:

In the details of each logon group you can define how many resources can be consumed on the instances within the logon group:

The predefined values are here:

This RFC logon group is then assigned in SMQS/SMQR to the scheduler:

On the sender’s side you have to define a RFC connection to the CRM with a logon group that exists in CRM: SM59.

Example:
Logon groups can be defined in transaction SMLG. See http://help.sap.com/saphelp_nw73ehp1/helpdata/en/c4/3a64b4505211d189550000e829fbbd/frameset.htm.

When there are several servers in the logon group and there are thousands of parallel RFC calls to CRM the load balancing may not be correct. The rules taken for load balancing are based on the statistics of the message server which are only updated every 5 minutes. See for this: http://help.sap.com/saphelp_nw73ehp1/helpdata/en/28/1c623c44696069e10000000a11405a/frameset.htm

In a situation where a large volume of bdocs are processing at the same time day to day work is occurring, (second phase cutover for example) this global design may not be ideal. More specific control at the individual queue level may be required so an advanced model must be used.

For some projects, data conversion loads will be occurring while normal day to day processing is occurring. There may be no scheduling or implementation way around this. This means that the majority of the dialog processes during this time will be used by a large volume of data conversion bdocs representing high volume data objects (i.e. IBases, Contracts, BP’s, etc.). Users will experience delays waiting for “their” bdocs to process because there are no dialog processes available.

The queue mechanism does not distinguish a high volume load bdoc from a normal user bdoc. With a standard configuration, there is no way to prioritize one bdoc over another or separate a user load from a data conversion load. We will now describe options to help prioritize load and manage queue execution.

By using specific queue names and the ‘userdest’ parameter we can trick objects into going in to the application server or even logon group that we desire.
qRFC communication is an extended form of tRFC communication. It adds a serialization concept to the transactional concept of tRFC. qRFC communication consists of tRFC communication with a preceding inbound queue and outbound queue. The ‘userdest’ parameter is actually a leftover from tRFC. It is possible to use this parameter to throttle queue loads with the correct setup.

When you register a queue, the data is stored in table QIWTAB. This table holds the information for the queue scheduler, server groups, user destination, and other parameters.

Table QIWTAB:
The queue scheduler is another special queue!

**Mapping Userdest**

‘Userdest’ parameter is an RFC connection defined in SM59.

A specific user can also be defined for each RFC destination. This is very good for later issue analysis.
The key here is that an individual RFC destination can have LOGON GROUP!
With logon groups we can throttle the load to individual servers or server groups.
qRFC Process Flow Summary with userdest:

Expected Result: Queue distribution based on logon group:

The next step is usage of Max Runtime Parameter Changes. This is a simple option with coarse control. The max runtime parameter controls how long the queue scheduler runs. In this case “run” means how long does it go and try to grab objects to process, based on the availability of system resources. It seems simple, but this allows some prioritization in queue execution.

For example:
If we set the max runtime for a user ibase queue to 60s and set the max runtime for large load ibase queue to 5s, the system will spend more time trying to process the user ibase queue. It will wait to process the current objects before grabbing the next 5s, or 60s worth of objects.
The next time the scheduler pulls, it will grab 60s worth of user ibase objects and only 5s of load ibase. So, as dialog processes become available, the system will try to fill dialog slots with more user ibase bdocs, rather than load ibase bdocs.

**Expected Result: Queue distribution based on max runtime:**

In large queuing systems this can get complicated. A balance must be found within prioritization, but unfortunately there is no hard and fast formula for this. Prioritization is highly dependent on how long each bdoc runs, available dialog at a given time, etc. Experimentation is the best approach here. Settings may change depending on the load type and queue being executed at the time. Documenting each load and run phase is the key to making this approach work.

In this context take a look at the definition of the parameter rdisp/rfc_check: rdisp/rfc_check is used to control the check as to whether sufficient dialog work processes are free for processing asynchronous RFC calls. When it sets to 1, system will monitor all the start of asynchronous RFCs. However, all RFCs that were started from asynchronous RFC sessions are not monitored. That's why you can see more DIA work processes are used by RFC during mass load.

Setting this parameter to 2 can help you restrict all RFCs that were started from asynchronous RFC sessions. However, you may need to increase the number of dialog processes that can be used for RFC to make sure RFC sessions will not be blocked by the
limitation on RFC resources. Please go through Note 74141 carefully regarding this parameter and how to do resource management for RFCs.

### 1.7.2.4 Housekeeping

In this sub-chapter we will give some overview of the most important to-dos to keep the system clean and help keeping the performance constant and the DB growth lower.

#### Org-Management buffering

The organization model can be stored in cluster tables via report HRBCI_ATTRIBUTES_BUFFER_UPDATE. This technique improves access to organizational structure, for example, the inheritance of the attributes does not need to be processed online. Background: Entries in the cluster tables are only valid for one day. Therefore, the cluster table has to be created on a daily basis by report HRBCI_ATTRIBUTES_BUFFER_UPDATE. Otherwise, performance improvements by this technique are not possible.

Set up the buffering indicator via transaction OOATTRCUST for the scenarios you are using. Afterwards, schedule a periodic daily running background job with the report HRBCI_ATTRIBUTES_BUFFER_UPDATE. This job should run after midnight every day. In case you do an HR structure upload from an HR system execute the report afterwards.

#### Application Log Size (BALHDR)

System wide performance can be impacted due to the large size of the application log tables, BALHDR etc. Reduce the number of entries in table BALHDR on a regular basis. Use transaction SLG2 to delete entries out of table BALHDR as described in SAP Note 195157.

#### Middleware tables SMW*_BDOC* can grow very large

Depending on the load of your trace, the middleware log tables can grow larger than 100 MB. Consequently, it can often be observed that expensive insert and read operation on the SMW3_BDOC* tables take place.

Take administrative actions to reduce the size of the tables according to SAP Note 206439. Only increase the trace level if necessary. Furthermore SAP Note 835761 has to be implemented to remove the BDocs in status I04. Schedule report SMO6_REORG2 on a daily basis.

If the table SRRELROLES contains many entries with the OBJTYPE TRANSID consider the SAP Note 138576 to reduce them.

If the table SRRELROLES contains too many entries with the OBJTYPE BUS* you have remove them by archiving the corresponding objects using the transaction SARA.

#### MIME Repository consistency check

After each SP import, execute the report : RSIR_MIME_EXPIRY_CLEANUP on node /SAP in order to avoid repetitive download of static mime objects from the server, which otherwise may be stored in the browser cache. Make sure Z-Style sheets have at least 1 week caching time defined in the P-System.
1.7.2.5  Performance tips and tricks
Check out the SDN for latest performance tips and tricks. There are also some good blogs on CRM performance around, e.g.

1.7.2.6  Enqueue Server
Enqueue Server is installed standalone:
http://help.sap.com/saphelp_nw73ehp1/helpdata/en/47/e0a62341744ac5e10000000a42189d/frameset.htm
If there is a bottleneck in ENQUEUE you can raise the number of threads up to four with this parameter: enque/server/threadcount
See also:
http://help.sap.com/saphelp_nw73ehp1/helpdata/en/47/ea3fd400e83b8be10000000a421937/frameset.htm
Check also SAP Note 959733.

Security enhancement available in CRM
In case the CRM WebClient is offered directly as a frontend to the dealers via the internet (without using a VPN):
Make logon service pages not scrollable by search engines. Add Meta-Tag
(<META NAME="ROBOTS" CONTENT="NONE">)
Check if Whitelist checks can help for external calls.
For CRM e.g. available via
Enhancement Spot Name: WCF.Restrict_ACCESS
BAAd Name: WCF.Restrict_ACCESS_BADI
Interface to be implemented: IF_WCF.Restrict_ACCESS_BADI
(BAAd: Restriction of Access to WebClient UI)
This Business Add-In (BAAd) is used in the component CA-WUI-APF. You can use this BAAd to restrict the access to major parts of the WebClient UI.
The BAAd uses the concept of white lists, that is, the BAAd implementation can return a list of allowed business roles and a list of allowed UI components. Then only these business roles and UI components are made available to the user.
The implementation reacts on the HTTP header field ‘x-sap-crm-external-client’. This field is intended to be set on the SAP Web Dispatcher for all incoming requests to the CRM system.
It is possible that the BAAd implementation is only executed for external requests and not for internal requests.