

**Billing to Cash Collection Project**

**Interface Detail Design Document**

*Development ID I046 – Contact Publish*

*IB Interfaces: Siebel CRM to SAP Contacts*

[1) Approvals 4](#_Toc259113393)

[2) Review List 4](#_Toc259113394)

[3) Related Documentation 4](#_Toc259113395)

[4) Glossary 4](#_Toc259113396)

[5) Introduction 5](#_Toc259113397)

[**5.1** **Document Context** 5](#_Toc259113398)

[**5.2** **Document Structure** 5](#_Toc259113399)

[**5.3** **Scope** 5](#_Toc259113400)

[**5.4** **Out of Scope** 5](#_Toc259113401)

[**5.5** **Assumptions** 5](#_Toc259113402)

[6) User Contacts 6](#_Toc259113403)

[**6.1** **Interface Definition** 6](#_Toc259113404)

[6.1.1 Interface Purpose 6](#_Toc259113405)

[6.1.2 Context 6](#_Toc259113406)

[6.1.3 Functional View 7](#_Toc259113407)

[6.1.4 Interface Characteristics 7](#_Toc259113408)

[**6.2** **Interface Design** 8](#_Toc259113409)

[6.2.1 High Level Interface Design 8](#_Toc259113410)

[6.2.2 Data Definitions 9](#_Toc259113412)

[**6.3** **Interface Design Components** 10](#_Toc259113413)

[8.3.1 CCRMCM\_DequeueContactRequest 10](#_Toc259113414)

[8.3.2 Contact Event Topic 11](#_Toc259113415)

[8.3.3 SAPCM\_EventManager 12](#_Toc259113416)

[8.3.4 SAPCM\_Invoker 14](#_Toc259113417)

[8.3.5 PL-SQL 15](#_Toc259113418)

[8.3.6 Data Mapping (Transformation) 15](#_Toc259113419)

[7) Developer Notes 15](#_Toc259113420)

[ **Code management** 15](#_Toc259113421)

[ **Deployment Process:** 15](#_Toc259113422)

[8) Issues/Workarounds: 16](#_Toc259113423)

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change Description | Author | Date Released |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Approvals

The individuals listed below will be required to review and approve this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
|  |  |  |  |
|  |  |  |  |

# Review List

|  |  |
| --- | --- |
| Name | Position |
|  |  |

# Related Documentation

| Ref | Title | Author | Version |
| --- | --- | --- | --- |
| I046 | User Contacts | Elash Morjaria | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Glossary

| Abbreviation | Definition |
| --- | --- |
| CCRM | Commerce and Customer Relationship Management |
| EAI | Enterprise Application Integration |
| B2CC | Bill To Cash Collection |
| SAP | SAP System |

# Introduction

##### **Document Context**

This document is one of a set of Interface Definition Documents providing the definition and High Level Design of the interfaces required for the Siebel CRM to SAP B2CC inbound interface that will enable the User contact creation, User contact modification, User swap or User Merge of Contacts within SAP B2CC, which will ultimately be driven by the Siebel CRM system.

##### **Document Structure**

This document describes about the requirements of an interface in terms of scope, functionality and characteristics. This document focus on what the interface does from an end-to-end perspective. Also covers the design of interface and the technical integration implementation.

##### **Scope**

This document covers the interface that will support the Enterprise Application Integration (EAI) solution by providing functionality to allow:

Siebel Order Management activity will dictate the behavior in which orders are processed in SAP B2CC. The process flow described in the following section depicts the subsequent processing of a Siebel CRM Order within SAP B2CC for the creation and amendment of a SAP user contact details

When billable order lines are released to billing within Siebel CRM, the order data is sent via the EAI for processing in SAP B2CC. The order lines will consist of contact details for user targeted products.

Any contact that is sent to SAP will be marked as SAP Published in Siebel and subsequent any updates to that contact will be sent to SAP.

##### **Out of Scope**

* NA-

##### **Assumptions**

* Siebel CRM will be the single point of entry for order processing. All Contacts must be raised, changed, or removed within Siebel CRM and subsequently fed to SAP B2CC.

# User Contacts

##### **Interface Definition**

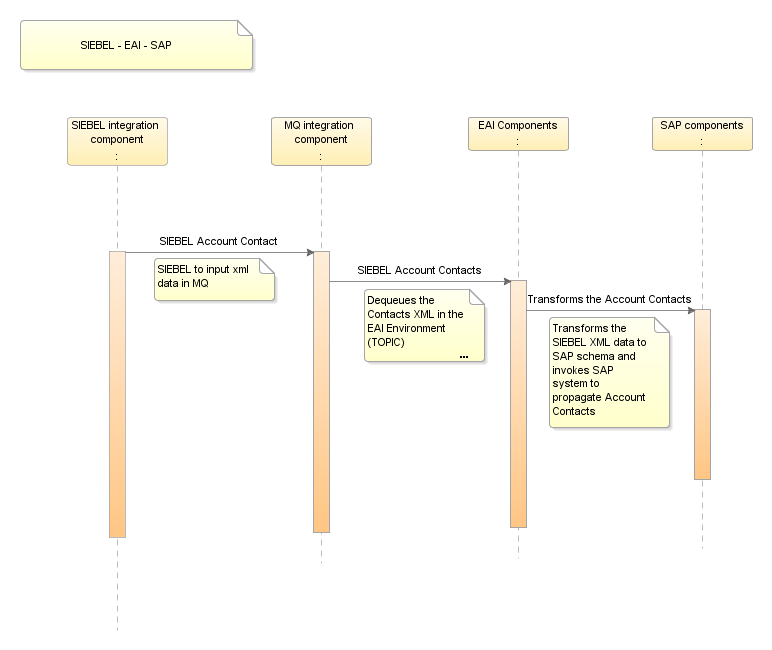
* 1. Interface Purpose

The purpose of this interface is to allow Siebel CRM to create, modify, merge, and swap user contacts in SAP-B2CC for processing.

|  |  |
| --- | --- |
| **Function** | **SAP-B2CC** |
| Creation of User Contact. | Yes |
| Modification of User Contact | Yes |
| Merge of User Contact | Yes |
| Swap of User Contact | Yes |
| Delete a Contact | Yes |

* 1. Context

The creation of User contacts process is initiated from Siebel CRM. This is part of an overall process which governs the sale or modification of the provision of Thomson Reuter’s services to customers. EAI will orchestrate the end to end process by invoking the creation of contact process on the SAP System through message-based event.

* 1. Functional View
  2. Interface Characteristics

EAI dequeues the Contact data from MQ e\_CRM\_EAI\_027\_CON\_PUB\_RQST\_Q to which Siebel enqueues request message. The BPEL process, CCRMCM\_DequeueContactRequest dequeues the Contact data. The Contact interface is offered by EAI to Siebel to allow Create, modify, Swap and Merge the valid contact data in SAP.

The messaging style used for the User contact interface is the ‘Command Style’, where a general purpose message format is used with a standard data payload and the required functionality is specified by an ‘action code’.

##### **Interface Design**

* 1. High Level Interface Design



NOTE: **BLUE**: Existing BPEL Components

* 1. Data Definitions

The message specifications used in the interfaces between EAI and the external systems are defined in the API specification [6].

Latest versions of the XML schemas for the messages used in the interface are stored in the Reuters Developer Network Subversion repository (https://int.thomsonreuters.com) project called sami (https://sami.cdt.int.thomsonreuters.com). The definitions for this interface can be found in subversion at:

https://sami.cdt.int.thomsonreuters.com/svn/eai\_eai/branches/6.1/design/EAI1.0/Interfaces

##### **Interface Design Components**

|  |  |  |
| --- | --- | --- |
| **Module Name/Method** | **Module Type** | **Existing/New** |
| CCRMCM\_DequeueContactRequest | BPEL Process | Existing BPEL Component |
| Contact Event Topic | JMS Topic | Existing Topic |
| SAPCM\_EventManager | BPEL Process | New BPEL Component |
| SAPCM\_Invoker | BPEL Process | New BPEL Component |

1. CCRMCM\_DequeueContactRequest

* **Purpose**

The BPEL process dequeues the User Contact request from the MQ and process for validating the data to create the contacts in SAP.

* **Program Logic**

**AS-IS Process:**

The process dequeues the message from the queue e\_CRM\_EAI\_027\_CON\_PUB\_RQST\_Q to which Siebel has enqueued. ‘e’ is the environment P- prod, B – blue, BL – Blue Lite, A – UAT, Q – System test, D – Support, C – Dev and NF- Non Functional.

This process transforms data into a canonical contact event (EAI.MSG.CM.197) and writes this to the ContactEventTopic.

As part of the transformation the date format is to be transformed to ‘YYYY-MM-DD’ and date time format to ‘YYYY-MM-DDTHH24:MI:SS’.

Siebel sends the requests with date in the format ‘MM/DD/YYYY’.

**TO-BE:**

There will be Input Siebel schema (dequeue) changes and added new elements in the schema transformations with output schema.

* **DVM**

--To be added after defining the schemas--

* **Data Mapping (Transformation)**

--To be added after defining the schemas--

* **Scope:**

ValidateXML

* **Exception Handling:**

For all the scopes in CCRMCM\_DequeueContactRequest catch all types of exceptions and throwing the fault message.

Selection Failure**:**

For any Selection Failure fault exception while transformation catches the exception and throwing remote fault exception and appending the fault message to the title.

remoteFault:

For any Remote fault exception while transformation catches the exception and throwing remote fault exception and appending the fault message to the title.

runtimeFault:

For any run time fault exception while transformation catch the exception and throwing run time fault exception and appending the fault message to the title.

bindFault:

For any data binding or data validations exceptions while transformation catch the exception and throwing remote fault exception and appending the fault message to the title.

NOTE: After Fault the BPEL instance is ready for Re-initiate, Manually user/Ops team will be re-initiating after all validations done.

* **Input/output Schemas:**

EAI.MS.CCRM.194.xsd

EAI.MS.CCRM.197.xsd

* **Partner links**

**ContactEventTopic.wsdl**

**DequeueContactRequest.wsdl**

1. Contact Event Topic

* **Purpose**

The contact event topic provides a first-in-first-out publish-subscribe capability so that contact updates can be propagated to multiple consuming systems independently.

* **Program Logic**

**AS-IS:**

Contact events are written to the topic by the publishing component. BPEL JMS adapters are used to write to Topic and consume from the topic.

**TO-BE:**

The SIEBEL schemas will be defined with additional fields, based on which the functional transformations will be implemented.

* **DVM**

-NONE-

* **Data Mapping (Transformation)**

The topic does not perform any data mapping or transformation.

* **Input/output Schemas:**

EAI.MSG.CCRM.CM.198\_ContactRequest.xsd

* **Partner links**

NA

1. SAPCM\_EventManager

* **Purpose**

Receives contact events and initiates calls to the SAP Invoker to update SAP based on the contact events. The SAPCM Contact Manager uses the EAI Event Sequencer so that multiple instances can be run concurrently whilst maintaining sequential integrity of updates that could potentially conflict with the other.

* **Program Logic:**
* The process receives a PublishContactEventRequest (EAI.MSG.SAP.CM.198), checks with the EAI Event Sequencer whether there is a current in-progress middleware process that might affect this contact and waits if necessary until the event sequencer notifies it that it can processed. This check is done by sending a CheckSequenceRequest (EAI.MSG.UTL.185) to the Event Sequencer and then waiting for a BasicResponse (EAI.MSG.EAI.169) before proceeding.
* The data from the Topic is read, transformed in the interface\_buffer table schema and then sent to the table using a database adapter.
* Also the sequence checker (SC) is invoked from the SAPCM\_EventManager and SC is initiated.
* The sequence is released in Invoker only upon completion of the process and the sequence is released for the corresponding request instance.
* Although the SAPCM Contact Manager is responsible for acquiring the lock with the EventSequencer, it is not responsible for releasing it. This is done by the SAPCM Invoker after it has completed processing the contact update.
* **Data Mapping (Transformations)**

Process passes the same message which has been received from SAPCM account event subscriber but with the original tracing ID replaced with a unique correlation ID generated by this process.

* **DVM:**
* Shall be defined after the schemas are shared --

* **Partner links:**

DequeueContact.wsdl – MQ Adapter,

SequenceChecker.wsdl.

InterfaceBuffer.wsdl –Database Adapter

* **Exception Handling:**

For All the scopes in SAPCM\_EventManager catch all types exceptions and throwing the fault message.

remoteFault:

For any Remote fault exception while transformation in Parallel flows catch the exception and throwing remote fault exception and appending the fault message to the title.

runtimeFault:

For any run time fault exception while transformation in Parallel flows catch the exception and throwing run time fault exception and appending the fault message to the title.

bindFault:

For any data binding or data validations exceptions while transformation in Parallel flows catch the exception and throwing remote fault exception and appending the fault message to the title.

NOTE: After Fault the BPEL instance is ready for Re-initiate, Manually user/Ops team will be re-initiating after all validations done.

1. SAPCM\_Invoker

* **Purpose**
* This process receives a general contact update event and calls the appropriate SAP web service based on the update type.
* **Program Logic:**

The SAPCM Invoker performs logic to prepare the contact details for SAP because the logical data model required by SAP is quite different to the logical model published by Siebel. We also use the action codes provided by Siebel to control routing logic and the action codes passed to SAP

SAPCM\_EventManager which reads Siebel set data from topic inserts the same data into the interface buffer table.

SAPCM\_Invoker polls data from SAPCM\_Polling\_V (View Created for Interface buffer) for all the messages with the interface\_id=’SAPCM’ , state set as ‘N’ or ‘R’ for every 10 seconds.

It then checks for any running instances which would affect the initiation of this instance. If found any, the process waits for the completion of the same and in the other case the process in initiated.

After the completion of the entire flow, SAPCM\_Invoker invokes the Sequence Releaser which releases the sequence.

The above process follows the update call to the interface buffer table. During which the state is changed to ‘C’ upon completion of process.

* **Exception Handling:**

For All the scopes in SAPCM\_ Event Manager catch all types exceptions and throwing the fault message.

remoteFault:

For any Remote fault exception while transformation in Parallel flows catch the exception and throwing remote fault exception and appending the fault message to the title.

runtimeFault:

For any run time fault exception while transformation in Parallel flows catch the exception and throwing run time fault exception and appending the fault message to the title.

bindFault:

For any data binding or data validations exceptions while transformation in Parallel flows catch the exception and throwing remote fault exception and appending the fault message to the title.

NOTE: After Fault the BPEL instance is ready for Re-initiate, Manually user/Ops team will be re-initiating after all validations done.

* **Data Mapping (Transformations)**

TransCreateContactsToSAP.xsl

* **Partner links:**

SAPCM\_POLLING (DB Adapter)

SAP\_Create\_Contacts.wsdl (SAP Provided WSDL i.e. external wsdl)

* **Database:**

Interface\_buffer,

Interface\_Config

1. PL-SQL

**SAPCM\_POLLING\_V**

CREATE OR REPLACE FORCE VIEW &topEAIOwnerUserName.."SAPCM\_POLLING\_V" ("ID", "CREATED", "PAYLOAD", "INTERFACE\_ID", "REFERENCE\_ID", "STATE", "LAST\_UPDATED") AS

select ib.id, ib.created, ib.payload, ib.interface\_id, ib.reference\_id, ib.state, ib.last\_updated

from eaiowner.interface\_buffer ib, eaiowner.interface\_config ic

where ic.ID = 'SAPCM'

and ic.state = 'ON'

and ic.IS\_AVAILABLE = 'Y'

and ib.INTERFACE\_ID = ic.ID

and ib.STATE in ('N','R')

and ib.SEND\_ATTEMPTS < ic.MAX\_SEND\_ATTEMPTS;

**SAP CM Interface Config:**

INSERT INTO INTERFACE\_CONFIG (ID, STATE, IS\_AVAILABLE, SHUTOFF\_THRESHOLD\_SECONDS, SHUTOFF\_THRESHOLD\_MESSAGES) VALUES ('SAPCM','ON','Y', NULL, 10);

1. Data Mapping (Transformation)

# Developer Notes

##### **Code management**

For Code management and usage of subversion refer the document.

##### **Deployment Process:**

The deployment process for BPEL, ESB or web services components and as well as for any other deployments like DB Scripts, Business rules configurations: refer EAI- Build and Deployment Process.doc in https://sami-crm6-eai.reutersdev.net/svn/sami-crm6-eai/trunk/dev/EAI-CRM61/R2releases/BuildDocs folder in SVN.

# Issues/Workarounds:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Summary | Status | Assigned to | Solution |
| 1 | Siebel CRM Schemas | Open | CRM team |  |
| 2 | SAP WSDLs | Close | SAP Team |  |
| 3 | EAI Dev environment details required | Close | EAI |  |