



# Online Interaction Interface (OII)

August 2019



# Revision History

Date	Description
24.4.2006	The original release version.
26.4.2006	Added pseudo queues OII_PHONE and OII_EMAIL.
5.10.2006	Added WicomFileAccess installation.
20.10.2006	Updated OII screenshot (removed test functions).
28.3.2007	Updated to Contact Center 5.5. Support for SAP CRM 5.0 added. Conference and dropParty operations added.
18.5.2007	Added into installation "Customizing file".
22.5.2007	Simplified installation: no more Customizing file, registry settings CEM_NAME and CEM_APP_NAME
28.5.2007	Added a picture of IIS 6.0 with ASP.NET v2.0.50727 allowed.
29.5.2007	Removed references to J2EE.
12.6.2007	New CEM application parameters: OII.IDD, OII.Country, OII.RemovePlusPrefix
26.6.2007	Added: Configuring OII to use Windows authentication
8.8.2007	Removed app.config, because registry is used by default.
17.9.2007	Added automatic detection of system configuration (application, queue) changes.



- 
- 30.10.2007** Added requirement for SMTP server (IIS). Removed application parameters appcnf\_auto\_pw\_time\_callout, appcnf\_auto\_pw\_time\_direct, appcnf\_do\_STATS, appcnf\_do\_NOTES.
- 
- 27.11.2007** Added OII.IgnoreOperDirect
- 
- 30.6.2008** Updated for SAP Contact Center 6.0 SP1: Chat and Action Item channels, Item Attached Data.
- 
- 28.8.2008** Added information about Action Item queues.
- 
- 11.3.2009** Added SAPphone interface for Contact Center 6 SP5.
- 
- 17.3.2009** Added CRM\_ICI\_TRACE for ICI troubleshooting. OII.GetUserPresenceEmail is disabled by default.
- 
- 28.8.2009** Added parameters OII.ConvertEmail, OII.ShowBusyStatus, OII.BusyStatusDuration for Contact Center 6 SP6.
- 
- 14.4.2010** Added parameters OII.IgnoreCdtOutbound for Contact Center 6 SP7 and OII.SendDTMF, OII.WrapUpOutbound, OII.ActionMultiSession, OII.EmailMultiSession for Contact Center 6 SP8.
- 
- 23.4.2010** Added OII.PdBeginPreview and OII.PdEndPreviewReject, which are supported with CRM 7.0 EhP1 only.
- 
- 30.4.2010** Added OII.ForwardAndEnd and OII.TransferAndEnd parameters.
- 
- 22.8.2011** Updated to Contact Center 7.
- 
- 2.4.2012** Added OII.CallRecordingCad parameter and sample CAD with CallRecording value.
-



---

<b>3.4.2012</b>	Added OII.RequestedWrapUp and OII.MapSubscribeUrl parameters.
<b>4.6.2012</b>	Added OII.WrapUpEndedFromCdt parameter for Contact Center 7 SP4.
<b>2.8.2012</b>	Added chapter for SAPphone server connection limits. New SAPphone server connection string option REG_COUNT.
<b>18.10.2012</b>	Added OII.SetCurrentQueues parameter.
<b>5.11.2012</b>	Added documentation for SAPphone: SAP .NET Connector 3.0.
<b>12.11.2012</b>	Added information about SIP_CALL_ID in CAD.
<b>22.11.2012</b>	Added OII.ReverseDnsSubscribeUrl parameter.
<b>19.12.2012</b>	Added information about Visual C++ 2005 Runtime.
<b>11.2.2013</b>	Added cmsPing and cmsHostId to IciUser.getAttributes. Added IciSystem.getCallRecordings operation.
<b>28.5.2013</b>	Added information about Visual C++ 2012 and 2010 Runtimes. Added new parameters in Contact Center 7 SP5: OII.SetExtraData, OII.CheckDuplicateActionItem
<b>30.5.2013</b>	Added action item routing attributes like TO_ADDRESS.
<b>3.6.2013</b>	Added CRMC_UI_PROFILE.
<b>3.10.2013</b>	Changed the ICI specification version.
<b>13.10.2013</b>	Changed AnswerInfo parameter name.

---



---

<b>10.1.2014</b>	Added a note about HAC not totally stopping OII instance when inactivated.
<b>2.4.2014</b>	Changed BCM to Contact Center. Added IciSystem.createCallBack function.
<b>8.4.2014</b>	Added updateCallBack and getCallBacks functions.
<b>14.10.2014</b>	Minor edits.
<b>28.10.2014</b>	SOAP_TIMEOUT_MS added.
<b>12.4.2016</b>	Added OII.UsePickupFolder parameter.
<b>19.8.2016</b>	Added more elements to QueueInfo type: open, avgWaitingTime, maxWaitingTime, busy, wrapUp, notReady, paused
<b>5.9.2016</b>	Added action item DELETE = X attribute handling to OII.CheckDuplicateActionItem parameter description.
<b>12.01.2018</b>	The document template changed
<b>10.9.2018</b>	Added OII.ShowPresenceProfiles parameter. Changed runtime to: Microsoft Visual C++ 2015 Redistributable Update 3
<b>26.10.2018</b>	OII.ConvertEmail = 0 by default.
<b>25.3.2019</b>	Added action item REQUIRED_AGENT, REQUIRED_AGENT_EXPIRY_TIME.

---



# Table of Contents

<b>1 Introduction</b> .....	<b>14</b>
1.1 Security .....	14
1.2 SOAP in ICI Interface.....	15
1.3 RFC in SAPphone Interface.....	16
<b>2 ARCHITECTURE OVERVIEW</b> .....	<b>17</b>
2.1 ICI Interface.....	17
<b>3 Installation</b> .....	<b>21</b>
3.1 Prerequisites .....	21
3.1.1 Visual C++ Runtime .....	22
3.1.1.1 Contact Center 7 SP5 and Newer .....	22
3.1.1.2 Contact Center 7 SP4 and Older .....	23
3.2 IA Setup .....	23
3.3 Windows Authentication.....	25
3.4 Testing Installation .....	26
3.4.1 OII.asmx.....	26
3.4.2 Default.aspx .....	28
3.4.3 Test.asmx .....	29
<b>4 Configuration</b> .....	<b>31</b>
4.1 Virtual Unit Registry Settings .....	31
4.2 OII Specific Registry Settings .....	32
4.3 OII Module Parameters .....	35
4.3.1 OII Module Parameters.....	36
4.3.2 Queue Parameters.....	49
4.3.3 Pseudo Queue .....	50
4.4 Notification Messages .....	50
4.4.1 OII.MapSubscribeUrl Parameter.....	51
4.4.2 OII.ReverseDnsSubscribeUrl Parameter.....	51
4.5 SOAP Proxy .....	52
4.6 Users .....	52
4.7 Queues.....	53
4.7.1 Enabling and Disabling .....	53
4.7.2 Informative Pop-Up Window .....	53
4.7.3 Action Item Queue .....	54
4.8 Automatic Detection of System Configuration Change .....	55
<b>5 Configuring IC on SAP CRM</b> .....	<b>56</b>
5.1 Create RFC Destination .....	56
5.2 Define Contact Center as a Communication System .....	57
5.3 Define Connections.....	59
5.4 Define Profiles.....	59
5.5 Assign System to Profile .....	60
5.6 Business Role Configuration.....	60
5.7 Business Role Customizations .....	61
5.8 Set Profile Parameter for User.....	61
<b>6 Configuring SAPphone on SAP CRM/ERP</b> .....	<b>63</b>
6.1 RFC Destination.....	63
6.2 SAPphone Server.....	65



- 6.3 SAPphone Administration ..... 65
- 6.4 Work Center Settings ..... 66
  - 6.4.1 User-Specific Settings..... 67
  - 6.4.2 SAPphone UI ..... 68
- 7 ICI Functionality Overview ..... 69**
  - 7.1 ICI Interfaces ..... 69
  - 7.2 Item Attached Data ..... 75
    - 7.2.1 Call Attached Data (CAD) ..... 75
    - 7.2.2 SIP Call ID ..... 77
    - 7.2.3 Chat Attached Data..... 77
    - 7.2.4 Item Attached Data in ICI 3.07 specification ..... 79
  - 7.3 Action Item Routing Attributes ..... 80
  - 7.4 Knocking Calls ..... 82
  - 7.5 E-Mail Transformations ..... 83
    - 7.5.1 Inbound Messages..... 83
    - 7.5.2 Outbound Messages..... 83
  - 7.6 SOAP Network Bandwidth ..... 84
    - a. Troubleshooting with CRM ICI Trace ..... 84
- 8 SAPphone Functionality Overview ..... 86**
  - 8.1 SAPphone Server Connections ..... 86
    - 8.1.1 SAP .NET Connector 2.0..... 86
    - 8.1.2 SAP .NET Connector 3.0..... 87
  - 8.2 SAPphone Client Connections..... 87
  - 8.3 SAPphone Functions ..... 88
    - 8.3.1 Call Control Functions..... 88
    - 8.3.2 Reporting Functions (Client Connection) ..... 89
    - 8.3.3 Functions Concerning Predictive Dialing / Power Dialing..... 90
    - 8.3.4 Functions for Call-Attached Data from 4.5B Onwards..... 90
    - 8.3.5 Functions for Call-Attached Data in Interface Version 4.00A ..... 90
    - 8.3.6 Functions for Monitoring External Components from Rel. 4.5B Onwards ..... 90
    - 8.3.7 Function for Statistical Data Upload ..... 91
- 9 ICI Interface Specification ..... 92**
  - 9.1 IciSystem Interface ..... 92
    - 9.1.1 exchangeProductInformation Operation ..... 92
      - 9.1.1.1 Definition ..... 92
      - 9.1.1.2 exchangeProductInformation Message ..... 92
      - 9.1.1.3 exchangeProductInformationResponse Message ..... 93
    - 9.1.2 getWorkcenterCapability Operation ..... 93
      - 9.1.2.1 Definition ..... 93
      - 9.1.2.2 getWorkcenterCapability Message ..... 94
      - 9.1.2.3 getWorkcenterCapabilityResponse Message..... 95
    - 9.1.3 getPresenceQueueInfo Operation ..... 95
      - 9.1.3.1 Definition ..... 95
      - 9.1.3.2 getPresenceQueueInfo Message ..... 96
      - 9.1.3.3 getPresenceQueueInfoResponse Message ..... 96
    - 9.1.4 getUserPresence Operation ..... 97
      - 9.1.4.1 Definition ..... 97
      - 9.1.4.2 getUserPresence Message ..... 98
      - 9.1.4.3 getUserPresenceResponse Message ..... 98
    - 9.1.5 getCallRecordings Operation..... 99
      - 9.1.5.1 Definition ..... 99



- 9.1.5.2 getCallRecordings Message ..... 100
- 9.1.5.3 getCallRecordingsResponse Message ..... 100
- 9.1.6 createCallBack Operation ..... 101
  - 9.1.6.1 Definition ..... 101
  - 9.1.6.2 extraData XML Format ..... 101
  - 9.1.6.3 createCallBack Message ..... 102
  - 9.1.6.4 createCallBackResponse Message ..... 102
- 9.1.7 updateCallBack Operation ..... 103
  - 9.1.7.1 Definition ..... 103
  - 9.1.7.2 updateCallBack Message ..... 103
  - 9.1.7.3 updateCallBackResponse Message ..... 104
- 9.1.8 getCallBacks Operation ..... 104
  - 9.1.8.1 Definition ..... 104
  - 9.1.8.2 getCallBacks Message ..... 104
  - 9.1.8.3 getCallBacksResponse Message ..... 105
- 9.2 IciUser Interface ..... 106
  - 9.2.1 getAttributes Operation ..... 106
    - 9.2.1.1 Definition ..... 106
    - 9.2.1.2 getAttributes Message ..... 106
    - 9.2.1.3 getAttributesResponse Message ..... 107
  - 9.2.2 setCurrentWorkmode Operation ..... 108
    - 9.2.2.1 Definition ..... 108
    - 9.2.2.2 setCurrentWorkmode Message ..... 108
    - 9.2.2.3 setCurrentWorkmodeResponse Message ..... 109
  - 9.2.3 setCurrentChannels Operation ..... 109
    - 9.2.3.1 Definition ..... 109
    - 9.2.3.2 setCurrentChannels Message ..... 109
    - 9.2.3.3 setCurrentChannelsResponse Message ..... 110
  - 9.2.4 setCurrentQueues Operation ..... 110
    - 9.2.4.1 Definition ..... 110
    - 9.2.4.2 setCurrentQueues Message ..... 110
    - 9.2.4.3 setCurrentQueuesResponse Message ..... 111
  - 9.2.5 setAddresses Operation ..... 111
    - 9.2.5.1 Definition ..... 111
    - 9.2.5.2 setAddresses Message ..... 111
    - 9.2.5.3 setAddressesResponse Message ..... 112
  - 9.2.6 subscribe Operation ..... 112
    - 9.2.6.1 Definition ..... 112
    - 9.2.6.2 subscribe Message ..... 113
    - 9.2.6.3 subscribeResponse Message ..... 113
  - 9.2.7 unsubscribe Operation ..... 113
    - 9.2.7.1 Definition ..... 113
    - 9.2.7.2 unsubscribe Message ..... 114
    - 9.2.7.3 unsubscribeResponse Message ..... 114
- 9.3 IciEvent Interface ..... 115
  - 9.3.1 userChanged Operation ..... 115
    - 9.3.1.1 Definition ..... 115
    - 9.3.1.2 userChanged Message ..... 115
    - 9.3.1.3 userChangedResponse Message ..... 116
  - 9.3.2 subscriptionEnded Operation ..... 117
    - 9.3.2.1 Definition ..... 117





- 9.3.2.2 subscriptionEnded Message ..... 117
- 9.3.2.3 subscriptionEndedResponse Message ..... 118
- 9.3.3 containerChanged Operation ..... 118
  - 9.3.3.1 Definition ..... 118
  - 9.3.3.2 containerChanged Message ..... 119
  - 9.3.3.3 containerChangedResponse Message ..... 119
- 9.3.4 phoneCallChanged Operation ..... 120
  - 9.3.4.1 Definition ..... 120
  - 9.3.4.2 phoneCallChanged Message ..... 120
  - 9.3.4.3 phoneCallChangedResponse Message ..... 121
- 9.3.5 messageChanged Operation ..... 121
  - 9.3.5.1 Definition ..... 121
  - 9.3.5.2 messageChanged Message ..... 121
  - 9.3.5.3 messageChangedResponse Message ..... 122
- 9.3.6 chatChanged Operation ..... 123
  - 9.3.6.1 Definition ..... 123
  - 9.3.6.2 chatChanged Message ..... 123
  - 9.3.6.3 chatChangedResponse Message ..... 124
- 9.3.7 chatNewPosting Operation ..... 124
  - 9.3.7.1 Definition ..... 124
  - 9.3.7.2 chatNewPosting Message ..... 124
  - 9.3.7.3 chatNewPostingResponse Message ..... 125
- 9.3.8 actionItemChanged Operation ..... 125
  - 9.3.8.1 Definition ..... 125
  - 9.3.8.2 actionItemChanged Message ..... 126
  - 9.3.8.3 actionItemChangedResponse Message ..... 127
  - 9.3.8.4 Sample Data ..... 128
- 9.4 Iciltem Interface ..... 132
  - 9.4.1 create Operation ..... 132
    - 9.4.1.1 Definition ..... 132
    - 9.4.1.2 create Message ..... 133
    - 9.4.1.3 createResponse Message ..... 133
  - 9.4.2 wrapUpRequired Operation ..... 134
    - 9.4.2.1 Definition ..... 134
    - 9.4.2.2 wrapUpRequired Message ..... 134
    - 9.4.2.3 wrapUpRequiredResponse Message ..... 135
  - 9.4.3 wrapUpEnded Operation ..... 135
    - 9.4.3.1 Definition ..... 135
    - 9.4.3.2 wrapUpEnded Message ..... 135
    - 9.4.3.3 wrapUpEndedResponse Message ..... 136
  - 9.4.4 accept operation ..... 136
    - 9.4.4.1 Definition ..... 136
    - 9.4.4.2 accept Message ..... 137
    - 9.4.4.3 acceptResponse Message ..... 137
  - 9.4.5 reroute Operation ..... 137
    - 9.4.5.1 Definition ..... 137
    - 9.4.5.2 reroute Message ..... 138
    - 9.4.5.3 rerouteResponse Message ..... 138
  - 9.4.6 forward Operation ..... 138
    - 9.4.6.1 Definition ..... 138
    - 9.4.6.2 forward Message ..... 139



- 9.4.6.3 forwardResponse Message ..... 139
- 9.4.7 getAttachedData Operation ..... 140
  - 9.4.7.1 Definition ..... 140
  - 9.4.7.2 getAttachedData Message ..... 140
  - 9.4.7.3 getAttachedDataResponse Message ..... 140
- 9.4.8 setAttachedData Operation..... 141
  - 9.4.8.1 Definition ..... 141
  - 9.4.8.2 setAttachedData Message ..... 141
  - 9.4.8.3 setAttachedDataResponse Message ..... 141
- 9.4.9 deleteAttachedData Operation ..... 142
  - 9.4.9.1 Definition ..... 142
  - 9.4.9.2 deleteAttachedData Message..... 142
  - 9.4.9.3 deleteAttachedDataResponse Message ..... 142
- 9.5 IciContainer Interface ..... 143
  - 9.5.1 subscribe Operation ..... 143
    - 9.5.1.1 Definition ..... 143
    - 9.5.1.2 subscribe Message ..... 143
    - 9.5.1.3 subscribeResponse Message ..... 144
  - 9.5.2 unsubscribe Operation ..... 144
    - 9.5.2.1 Definition ..... 144
    - 9.5.2.2 unsubscribe Message ..... 145
    - 9.5.2.3 unsubscribeResponse Message ..... 145
- 9.6 IciPhoneLine Interface ..... 146
  - 9.6.1 getCalls Operation ..... 146
    - 9.6.1.1 Definition ..... 146
    - 9.6.1.2 getCalls Message ..... 146
    - 9.6.1.3 getCallsResponse Message ..... 146
  - 9.6.2 dropCalls Operation ..... 147
    - 9.6.2.1 Definition ..... 147
    - 9.6.2.2 dropCalls Message ..... 147
    - 9.6.2.3 dropCallsResponse Message ..... 148
- 9.7 IciPhoneCall Interface ..... 148
  - 9.7.1 getAttributes Operation ..... 148
    - 9.7.1.1 Definition ..... 148
    - 9.7.1.2 getAttributes Message ..... 148
    - 9.7.1.3 getAttributesResponse Message ..... 149
  - 9.7.2 dial Operation..... 149
    - 9.7.2.1 Definition ..... 149
    - 9.7.2.2 dial Message ..... 150
    - 9.7.2.3 dialResponse Message ..... 150
  - 9.7.3 consult Operation ..... 151
    - 9.7.3.1 Definition ..... 151
    - 9.7.3.2 consult Message ..... 151
    - 9.7.3.3 consultResponse Message ..... 152
  - 9.7.4 transfer Operation ..... 152
    - 9.7.4.1 Definition ..... 152
    - 9.7.4.2 transfer Message ..... 153
    - 9.7.4.3 transferResponse Message ..... 153
  - 9.7.5 reconnect Operation ..... 153
    - 9.7.5.1 Definition ..... 153
    - 9.7.5.2 reconnect Message ..... 154



- 9.7.5.3 reconnectResponse Message ..... 154
- 9.7.6 drop Operation ..... 154
  - 9.7.6.1 Definition ..... 154
  - 9.7.6.2 drop message ..... 155
  - 9.7.6.3 dropResponse Message ..... 155
- 9.7.7 hold Operation ..... 155
  - 9.7.7.1 Definition ..... 155
  - 9.7.7.2 hold Message ..... 156
  - 9.7.7.3 holdResponse Message ..... 156
- 9.7.8 unhold Operation ..... 156
  - 9.7.8.1 Definition ..... 156
  - 9.7.8.2 unhold Message ..... 157
  - 9.7.8.3 unholdResponse Message ..... 157
- 9.7.9 alternate Operation ..... 157
  - 9.7.9.1 Definition ..... 157
  - 9.7.9.2 alternate Message ..... 158
  - 9.7.9.3 alternateResponse Message ..... 158
- 9.7.10 conference Operation ..... 159
  - 9.7.10.1 Definition ..... 159
  - 9.7.10.2 conference Message ..... 159
  - 9.7.10.3 conferenceResponse Message ..... 159
- 9.7.11 dropParty Operation ..... 160
  - 9.7.11.1 Definition ..... 160
  - 9.7.11.2 dropParty Message ..... 160
  - 9.7.11.3 dropPartyResponse Message ..... 161
- 9.8 IciFolder Interface ..... 161
  - 9.8.1 getMessages Operation ..... 161
    - 9.8.1.1 Definition ..... 161
    - 9.8.1.2 getMessages Message ..... 161
    - 9.8.1.3 getMessagesResponse Message ..... 162
- 9.9 IciMessage Interface ..... 162
  - 9.9.1 getAttributes Operation ..... 162
    - 9.9.1.1 Definition ..... 162
    - 9.9.1.2 getAttributes Message ..... 163
    - 9.9.1.3 getAttributesResponse Message ..... 163
  - 9.9.2 getContent Operation ..... 164
    - 9.9.2.1 Definition ..... 164
    - 9.9.2.2 getContent Message ..... 164
    - 9.9.2.3 getContentResponse Message ..... 165
  - 9.9.3 setContent Operation ..... 165
    - 9.9.3.1 Definition ..... 165
    - 9.9.3.2 setContent Message ..... 165
    - 9.9.3.3 setContentResponse Message ..... 166
  - 9.9.4 delete Operation ..... 166
    - 9.9.4.1 Definition ..... 166
    - 9.9.4.2 delete Message ..... 166
    - 9.9.4.3 deleteResponse Message ..... 167
  - 9.9.5 send Operation ..... 167
    - 9.9.5.1 Definition ..... 167
    - 9.9.5.2 send Message ..... 167
    - 9.9.5.3 sendResponse Message ..... 168



- 9.9.6 setActionFlags Operation..... 168
  - 9.9.6.1 Definition ..... 168
  - 9.9.6.2 setActionFlags Message..... 168
  - 9.9.6.3 setActionFlagsResponse Message ..... 169
- 9.10 IciChatLine Interface ..... 169
  - 9.10.1 getSessions Operation..... 169
    - 9.10.1.1 Definition ..... 169
    - 9.10.1.2 getSessions Message..... 170
    - 9.10.1.3 getSessionsResponse Message ..... 170
- 9.11 IciChatSession Interface ..... 171
  - 9.11.1 getAttributes Operation ..... 171
    - 9.11.1.1 Definition ..... 171
    - 9.11.1.2 getAttributes Message ..... 171
    - 9.11.1.3 getAttributesResponse Message ..... 172
  - 9.11.2 invite Operation..... 172
    - 9.11.2.1 Definition ..... 172
    - 9.11.2.2 invite Message ..... 173
    - 9.11.2.3 inviteResponse Message..... 173
  - 9.11.3 leave Operation..... 173
    - 9.11.3.1 Definition ..... 173
    - 9.11.3.2 leave Message..... 174
    - 9.11.3.3 leaveResponse Message ..... 174
  - 9.11.4 getDialog Operation ..... 175
    - 9.11.4.1 Definition ..... 175
    - 9.11.4.2 getDialog Message ..... 175
    - 9.11.4.3 getDialogResponse Message..... 175
  - 9.11.5 postMessage Operation..... 176
    - 9.11.5.1 Definition ..... 176
    - 9.11.5.2 postMessage Message ..... 176
    - 9.11.5.3 postMessageResponse Message ..... 177
- 9.12 IciActionItem Interface ..... 177
  - 9.12.1 route Operation ..... 177
    - 9.12.1.1 Definition ..... 177
    - 9.12.1.2 route Message ..... 177
    - 9.12.1.3 routeResponse Message..... 178
    - 9.12.1.4 Sample Data ..... 179
- 10 ICI Content Description ..... 181**
  - 10.1 TextElement Element..... 181
    - 10.1.1 Definition ..... 181
    - 10.1.2 XML Schema Description ..... 181
  - 10.2 ProductInformation Element ..... 181
    - 10.2.1 Definition ..... 181
    - 10.2.2 XML Schema Description ..... 182
  - 10.3 QueueInfo Element ..... 182
    - 10.3.1 Definition ..... 182
    - 10.3.2 XML Schema Description ..... 183
  - 10.4 GroupInfo Element..... 184
    - 10.4.1 Definition ..... 184
    - 10.4.2 XML Schema Description ..... 184
  - 10.5 UserPresenceInfo Element ..... 184
    - 10.5.1 Definition ..... 184



10.5.2 XML Schema Description .....	185
<b>10.6 CollaborationData Element .....</b>	<b>185</b>
10.6.1 Definition .....	185
10.6.2 XML Schema Description .....	186
<b>10.7 CallRecording Element .....</b>	<b>186</b>
10.7.1 Definition .....	186
10.7.2 XML Schema Description .....	187
<b>10.8 CallBack Element.....</b>	<b>187</b>
10.8.1 Definition .....	187
10.8.2 XML Schema Description .....	188
<b>10.9 Address Element.....</b>	<b>189</b>
10.9.1 Definition .....	189
10.9.2 XML Schema Description .....	189
<b>10.10 User Element .....</b>	<b>189</b>
10.10.1 Definition .....	189
10.10.2 XML Schema Description .....	191
<b>10.11 Item Element.....</b>	<b>191</b>
10.11.1 Definition .....	191
10.11.2 XML Schema Description .....	192
<b>10.12 PhoneCall Element .....</b>	<b>192</b>
10.12.1 Definition .....	192
10.12.2 Phonecall Capabilities .....	193
10.12.3 Phonecall Status .....	194
10.12.4 XML Schema Description .....	195
<b>10.13 Message Element .....</b>	<b>195</b>
10.13.1 Definition .....	195
10.13.2 Message Capabilities.....	197
10.13.3 Message Status .....	197
10.13.4 XML Schema Description .....	198
<b>10.14 ChatSession Element .....</b>	<b>198</b>
10.14.1 Definition .....	198
10.14.2 ChatSession Capabilities .....	199
10.14.3 ChatSession Status .....	200
10.14.4 XML Schema Description .....	200
<b>10.15 ChatPosting Element .....</b>	<b>201</b>
10.15.1 Definition .....	201
10.15.2 XML Schema Description .....	201
<b>10.16 ActionSession Element .....</b>	<b>202</b>
10.16.1 Definition .....	202
10.16.2 ActionSession Capabilities .....	203
10.16.3 ActionSession Status .....	203
10.16.4 XML Schema Description .....	204
<b>10.17 ActionItem Element.....</b>	<b>204</b>
10.17.1 Definition .....	204
10.17.2 XML Schema Description .....	205
<b>10.18 Attribute Element .....</b>	<b>206</b>
10.18.1 Definition .....	206
10.18.2 XML Schema Description .....	206
<b>11 ICI WSDL Description .....</b>	<b>207</b>
<b>12 Troubleshooting.....</b>	<b>209</b>
<b>13 Glossary.....</b>	<b>212</b>





# 1 Introduction

This document describes *Online Interaction Interface (OII)* of the Contact Center software (known as SAP Business Communications Management (BCM) before software version 7 SP07). This information is directed to systems integrators and third-party software vendors who wish to interact with the telephony, e-mail and chat channel functions in a Contact Center system. The OII interface is dedicated to communicating with the ICI interface of the SAP CRM system, but it can be applied to other interfaces as well.

This document and the interface are subject to change without prior notice. The basic principles of the interface (such as use of SOAP over HTTP) are not subject to change.

Starting from version Contact Center 6.0 SP5, in addition to SOAP based ICI interface, OII supports also the SAPphone interface. SAPphone uses RFC (Remote Function Call), technology that can be used to call and execute functions or procedures on remote SAP systems or other computers.

## **NOTE:**

This specification applies to Contact Center 7.0, and newer, versions. If you are integrating BCM 6, or older, refer to the corresponding version specification.

## 1.1 Security

The interface provides a powerful tool to browse and manipulate data in the Contact Center system, and therefore you should protect the interface carefully from malicious use. Since the interface does not contain any authentication or authorization features, protect it with IP-level access restrictions or within the IIS configuration. Detailed information about configuring the service safely on the IIS server is available on request.



## 1.2 SOAP in ICI Interface

The interface is based on the SOAP version 1.1. The interface uses the HTTP protocol to carry SOAP messages between the SOAP client and itself.

If an operation fails for some reason, a standard SOAP 1.1 fault message is returned. The detail element contains an ICI SoapException.

```
<xs:complexType name="SoapException">
  <xs:sequence>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="areaCode" type="xs:int"/>
    <xs:element name="errorCode" type="xs:int"/>
    <xs:element name="errorText" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="errorRetry" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

```
HTTP/1.0 200 OK
Server: WWS 0.1
Date: Wed, 17 Sep 2003 21:30:59 GMT
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-type: text/xml
Content-Length: 238

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>com.sap.bcb.common.S SoapException</faultstring>
      <detail>
        <ns1:IciErrorCodes xmlns:ns1="urn:IciEventInterface">
          <areaCode>5</areaCode>
          <errorCode>2</errorCode>
          <errorText>SAP BCB:Other characters than +1234567890 in phone
number</errorText>
          <errorRetry>>false</errorRetry>
        </ns1:IciErrorCodes>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```





## 1.3 RFC in SAPphone Interface

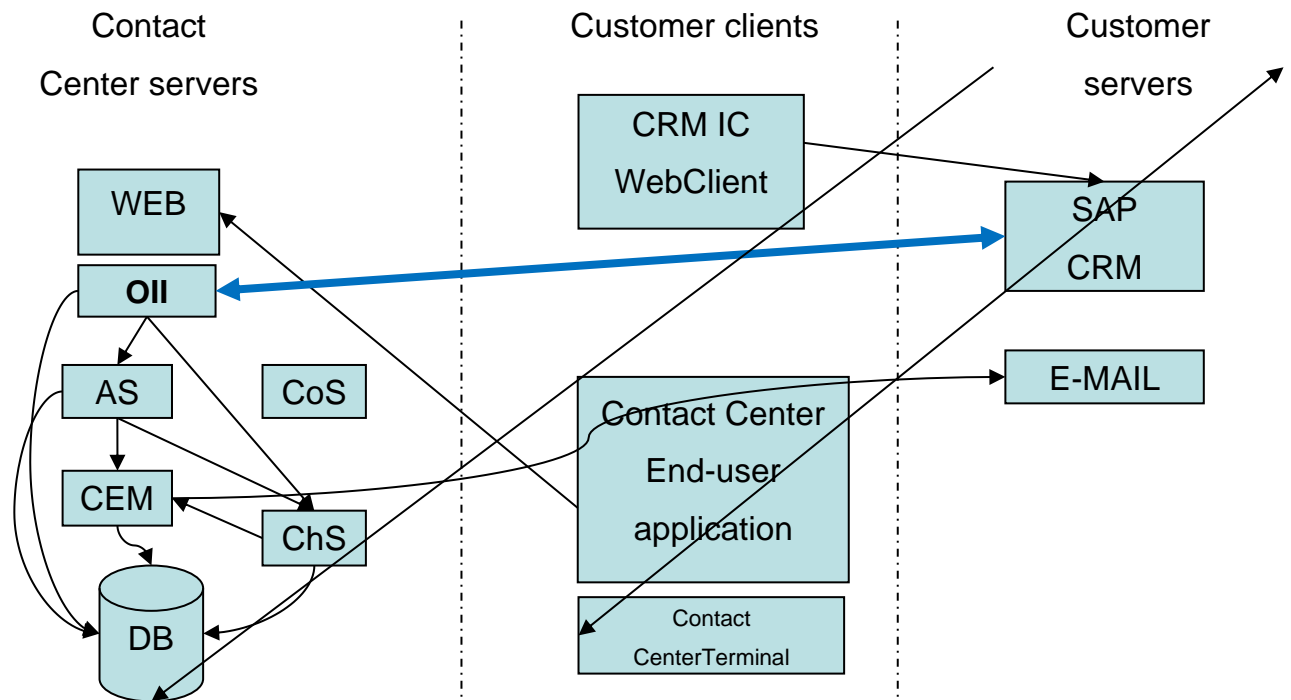
SAPphone interface is based on SAPphone specification 5.11.A11 (Jan 10, 2008). OII acts as both RFC server and RFC client. Most of the SAPphone RFC functions are implemented in OII as server functions, like SPS\_MAKE\_CALL, in which SAP CRM/ERP wants Contact Center to make a new outbound call. When there is a new incoming call, Contact Center wants to inform CRM/ERP, and OII calls client function SPS\_NEW\_CALL. So there are usually two connections, the server connection is mandatory and the client connection is optional. For more information about RFC, see SAP RFC documentation.



# 2 ARCHITECTURE OVERVIEW

## 2.1 ICI Interface

The interface is a server-side component running under a web server (IIS ASP.NET). It connects on the Contact Center side to the Agent Server (AS) and Chat Server (ChS) via TCP/IP, to the databases by using ADO.NET, and to the ICI interface on the CRM side via SOAP over HTTP.



The OII interface makes it possible for a third party to control telephony functions in real time: to answer, make, and hang-up calls. The CDT application behaves as if the user had used the functions in the CDT user interface, though it does not need to be visible at all. For example, it may be hidden behind another user interface (IC WebClient).



The integration is enabled when the user is subscribed to OII using the same user ID (login name in Contact Center, not case-sensitive) as in the CDT application. In a Contact Center system, it is possible that integration is enabled in some CDTs, and not in the others. Furthermore, it is possible to use CDT normally when the integration is enabled. New calls made directly in CDT are not visible in the integration, but if a queue call is answered in CDT, it behaves as if the call was answered via the integration.

OII also implements the e-mail channel as ICI requires. E-mail channel can be used for processing new tasks that have been created from inbound e-mails. The interface supports only the tasks that have been created by CEM from e-mail channel, but you can use the Task Management Interface (TMI) for processing other tasks and creating new tasks. OII supports simultaneous task processing with the CDT application. It is possible to pick tasks from hunt-group queue and get an alert about this via OII.

Similar to e-mails, OII also supports action items that CRM creates using OII. Action items are allocated by CEM in same way as e-mails, but they don't have the normal e-mail subject and body text, so they should be handled with IC WebClient instead of CDT. Action items go to special ACTION queue by default.

The fourth supported channel is chat. Customer chat comes via Chat Portal to chat queue, from where CEM allocates the chat to some agent. Chat messages are shown in CDT and IC WebClient simultaneously.

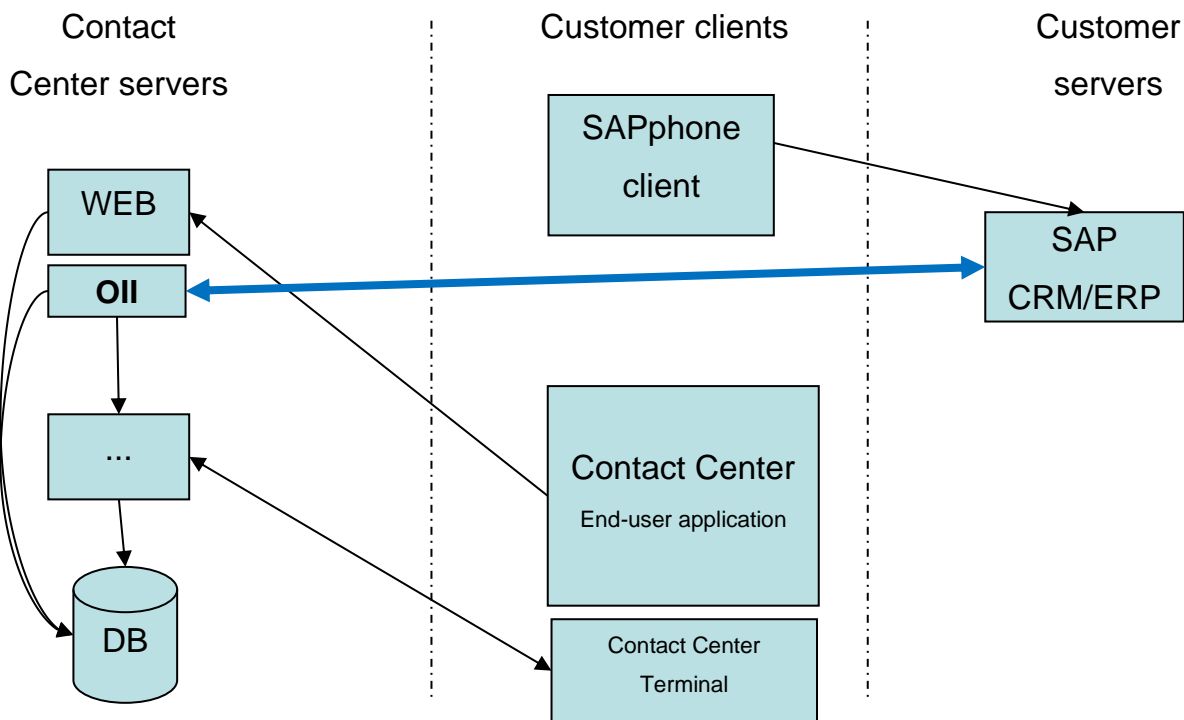
The interface offers only one phone call, e-mail message, action item, or chat session to a user at a time by default. Direct calls from agent to agent are not processed by default (except consultation calls), and they must be handled by using CDT.

OII is also used for Instant Messaging (IM) in CRM 7. IM is direct agent-to-agent chat. IM is similar to customer chat (which comes via Chat Portal), except that IM is independent of agent serving status, so agent may be having a phone call when IM is activated. IM is excluded from Contact Center reporting/monitoring.



### SAPphone Interface

Starting from the Contact Center 6.0 SP5, OII implements also SAPphone interface in addition to ICI interface. Because ICI requires that OII runs under the web server, also SAPphone interface is run there, even though no web server functionality is used. SAPphone interface can run concurrently with ICI interface. SAPphone interface creates two TCP/IP connections with CRM/ERP, server (mandatory) and client (optional).



SAPphone interface offers basic telephony support, a subset of ICI interface. No e-mail or chat is supported. SAPphone interface does not allow phone call state change notifications, so the SAPphone client may not know when a phone call ends, unless the agent drops the call using SAPphone function SPS\_DROPALL.



New incoming calls may be notified using a pop-up window (SAPphone UI starts). To enable incoming call popup, the agent must register using the SPS\_REGISTER, and there the EXT parameter must match the account number in Contact Center.

The logic in SAPphone interface is the same as in ICI interface, for example, by default only queue calls are alerted. Most OII parameters affect both ICI and SAPphone interface.



## 3 Installation

The interface is installed as part of the Integration Interfaces package in the Contact Center software. Install the package with the IA tool in an appropriate virtual unit.

### NOTE

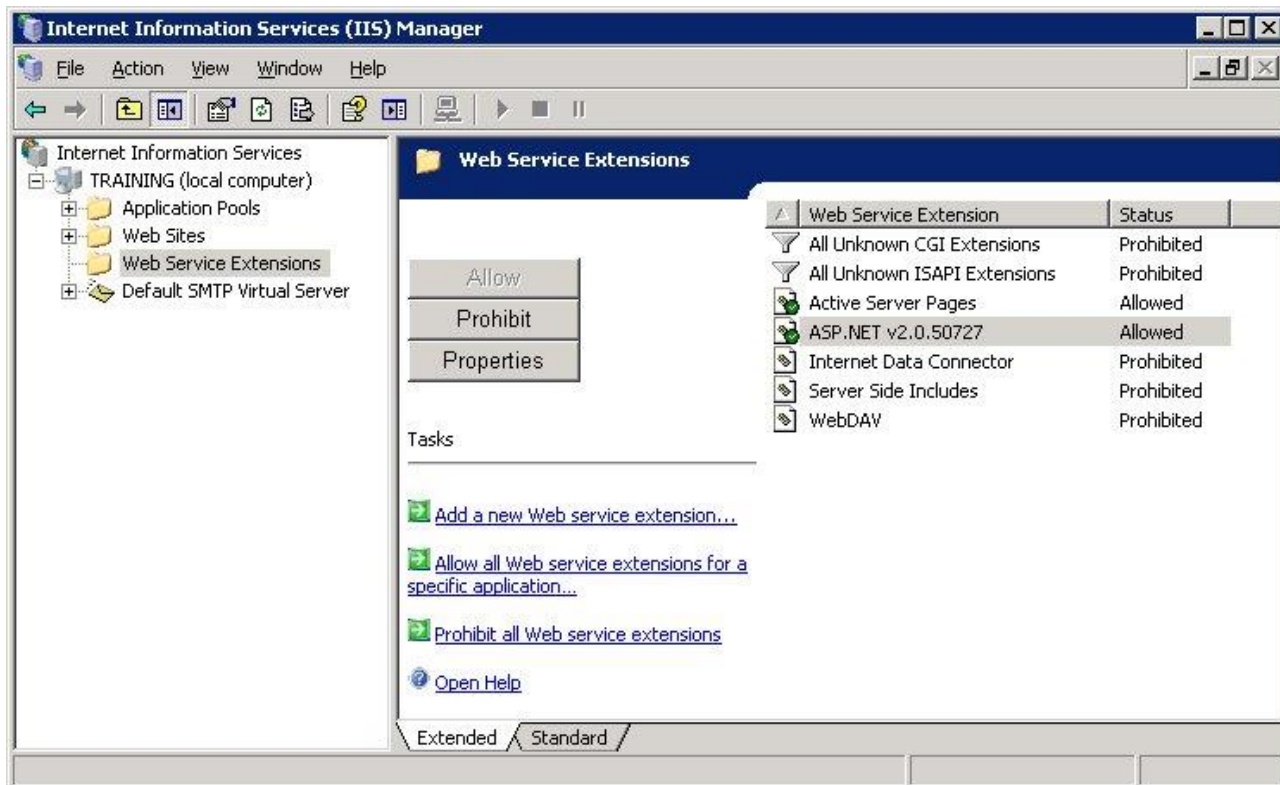
With software versions prior to SP08 it is not possible to switch off OII completely by inactivating it in IA but OII will start when a CRM user starts IC WebClient. To avoid this:

- Stop the application pool that OII is using in Windows, or make a copy of the application pool and assign it to OII, and keep that new application pool closed.
- Change the OII instance in IA also to Unassigned, because HAC cannot control the web application, when application pool is closed.

Remember to start the application pool and assign the OII instance to HAC after the issue is over, otherwise HAC cannot control OII after reactivation.

### 3.1 Prerequisites

The server where Integration Interfaces package will be installed needs to have the IIS version 5.1 (or later) and .NET Framework 2.0. ASP.NET v2.0.50727 must be allowed in IIS 6.0.



If the e-mail channel is used, install the SMTP Service subcomponent in the IIS, otherwise e-mail sending does not work. By default, OII uses the folder C:\Inetpub\mailroot for sending e-mails, it writes .eml file to subfolder Pickup, and the SMTP Server takes it from there. This folder location can be changed by the MAIL\_ROOT registry setting.

Starting from Contact Center 1608, by default the MessageQueue table is used instead of C:\Inetpub\mailroot\Pickup folder for sending e-mails. To use the Pickup folder, database module parameter OII.UsePickupFolder needs to be enabled (value = 1).

### 3.1.1 Visual C++ Runtime

#### 3.1.1.1 Contact Center 7 SP5 and Newer

OII requires the latest Visual C++ 2015 Runtime to be installed (32-bit). This can be usually installed from Windows Update. The latest version currently is:



<https://www.microsoft.com/en-us/download/details.aspx?id=53840>

Microsoft Visual C++ 2015 Redistributable Update 3 (x86)

vc redistrib.x86.exe

OII also requires Visual C++ 2010 Runtime for SAPphone integration:

<http://www.microsoft.com/en-us/download/details.aspx?id=5555>

Microsoft Visual C++ 2010 Redistributable Package (x86)

vc redistrib\_x86.exe

### 3.1.1.2 Contact Center 7 SP4 and Older

Older OII requires the latest Visual C++ 2005 Runtime to be installed (32-bit). This can be usually installed from Windows Update. The latest version currently is:

<http://www.microsoft.com/en-us/download/details.aspx?id=26347>

Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package MFC Security Update

## 3.2 IA Setup

In Contact Center Infrastructure Administrator (IA), OII is installed as part of Integration Interfaces. OII uses Configuration, Operative, Directory and Outbound database settings.

OII specific variables are saved to registry under OII, or to OII module parameter. See chapter 4 Configuration for more information.

Variable	Registry
OII_PORT	IpPort





Variable	Module parameter
OII_PORT	IpPort
OII.Email	OII.Email
...	See Configuration for the full list of module parameters.

The screenshot displays the SAP NetWeaver Administration Console interface. The left-hand pane shows a hierarchical tree view of the system components, with 'Integration Interfaces (7.0.1.10033)' selected under the 'Software' category. The right-hand pane is titled 'Element Properties: Integration Interfaces (7.0.1.10033)' and contains two main sections: 'Information' and 'Common Variables'. The 'Information' section lists details such as Software (Integration Interfaces), Description (SAP BCM Integration Interfaces), Install Media Path (D:\Program Files\SAP\OCM\Install), Base Installation (7.0.1.10033), and Components (IP 10.66.193.43 (7.0.1.10033) and \$VU\_NAME\$ WS OI (7.0.1.10033)). The 'Common Variables' section is a form with various configuration fields, including IP Address of the Virtual Unit (10.66.193.43), Log File Directory of the Virtual Unit (\$VU\_HOME\$\logs), Web Site Name (\$VU\_NAME\$), and several database connection parameters (Configuration Database Server Address or Name, Configuration Database Name, Configuration Database User Name, Configuration Database Password, Operative Database Server Address or Name, Operative Database Name, Operative Database User Name, Operative Database Password, TCP Port Number for HTTP (1080), TCP Port Number for HTTPS (1081), and Windows User Account for Running Web-Related Processes (DEVDFIAS406\comadmin)).



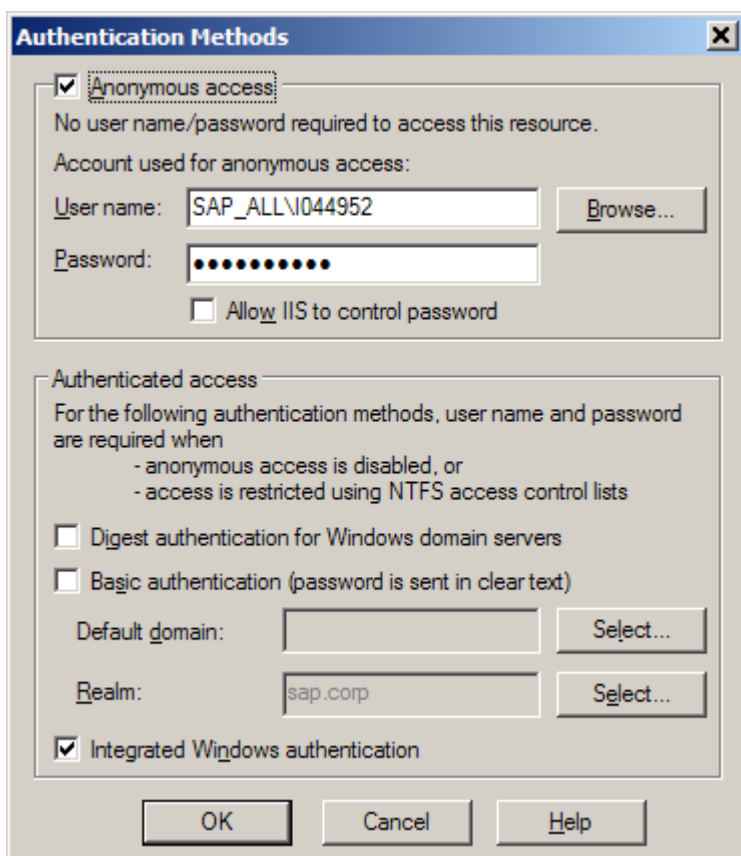
### 3.3 Windows Authentication

OII uses Windows authentication to access Configuration, Operative, Directory and Outbound databases (using VU registry settings CONFIGURATION\_DSN, OPERATIVE\_DSN, DIRECTORY\_DSN, OUTBOUND\_DSN), if UseWindowsAuthentication VU registry setting is "yes" or if SQL Server UID is not given. In order for this to work, Web.config must contain

```
<identity impersonate="true"/>
```

under <system.web>. This is done automatically by setup.

The UID used in Windows authentication is defined in IIS / OII / Properties / Directory Security / Anonymous access and authentication control / Edit / Anonymous access:



Database and file system access is done by ASPNET account (<machinename>\ASPNET), if Windows authentication is used. Installation adds Change (write) access to the C:\inetpub\mailroot\Pickup folder.



Starting from Contact Center 1608, by default the `MessageQueue` table is used instead of `C:\Inetpub\mailroot\Pickup` folder.

## 3.4 Testing Installation

### 3.4.1 OII.asmx

After installation, enter the web page: <http://ip-address:port/OII/OII.asmx>

## OII

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [EnumClients](#)
- [EnumItems](#)
- [EnumUsers](#)
- [GetHttpRequestTime](#)
- [GetLogPath](#)
- [GetState](#)
- [SetLogLevel](#)
- [Start](#)
- [Stop](#)
- [Unload](#)
- [UpdateSettings](#)

This page contains the following functions, which can only be executed if the web page has been loaded locally in the web server:

EnumClients	Enumerates currently active phone clients (CDTs).
-------------	---------------------------------------------------



EnumItems	Enumerates phone calls, e-mails and chats currently in OII memory.
EnumUsers	Enumerates users cached in OII memory.
GetHttpRequest	Returns information about the web site address and physical path of OII:  AppDomainAppId=/LM/W3SVC/2/ROOT/OII  AppDomainAppPath=D:\Program Files\SAP\BCM\VU\VU_OII\web\OII  AppDomainAppVirtualPath=/OII
GetLogPath	Returns the file path for the OII log file. It is in the same directory as other log files (for example, C:\Program files\SAP\BCM\VU\VU1\logs) and its name has the following syntax: OII_VU1_20050105.log.
GetState	Returns the Connected value if everything is OK. This means that OII has a connection to Agent Server.
Start	Starts the OII web service. The service is already started when you enter the page, or if the integration calls this page, but this function starts it again if it has been stopped.
Stop	Stops the integration. You must use the Start function to restart it.
Unload	Unloads OII web service from the memory (so that new settings can be read from registry). OII restarts



	automatically when any of the web pages are loaded (by the integration). The other way to unload OII is to modify web.config file (just saving it in Notepad is enough).
UpdateSettings	Makes OII to re-read the configuration settings (such as the queues) from the database . This is done automatically every midnight.

To ensure that OII works properly, configure the interface and Agent Server (AS), see the chapter [Configuration](#). Investigate the log file, and make sure it does not show any errors or exceptions.

OII restarts itself automatically once a minute until it gets a connection to AS and detects that the configuration is OK. To disable the automatic restarting, call the Stop function.

OII is not removed from memory when the Stop function is called. To stop OII and remove it from the memory, call the Unload function or modify the Web.config file (in C:\Program files\SAP\BCM\VU\VU1\web\OII). Just changing the modification date is enough.

### 3.4.2 Default.aspx

Web page: <http://ip-address:port/OII/default.aspx> displays the overall status of OII.



**SAP BCM OII Status**

Title	SAP BCM OII {Wicom_build}
Version	1.0.4104.17223
Domain	/LM/W3SVC/2/ROOT/OII_70-27-129457712556056634
VU	VU7
Application state	Connected
Chat Server state	Not connected
Objects	Clients:0 Items:0 Users:2 Queues:7 Subs:0
Threads	[SOAP:0/0/0/25 q=0/0 ms=0/0/0 #=0] [Chat:0/0/0/25 q=0/0 ms=0/0/0]
Server time	2011-03-28 13:59:43.445
Server UTC time	2011-03-28 10:59:43.445
Log file	C:\Program Files (x86)\Wicom\VU\VU7\logs\OII_70_VU7_20110328.log
Log level	TRC (3)
Log statistics	EXC:0 ERR:0 INF:187 TRC:515 DBG:7
Secure log	False
Last Err	
Last Exc	

This page displays the OII version information, Virtual Unit name, Application state, and so on. It also shows the current log file name, and the last logged error and exception if there are any.

Normally Application state should be Connected (connection with Agent Server), and also Chat Server state should be Connected.

The page is updated automatically once a minute. Displaying this page does not start OII, on the contrary to OII.aspx page.

### 3.4.3 Test.aspx

Test.aspx is similar to OII.aspx, but it just contains various testing functions. Not to be used externally.





## 4 Configuration

### 4.1 Virtual Unit Registry Settings

The following variables are set during installation of the Integration Interfaces package with the IA tool, to VU level into registry:

Setting	Default value	Description
CONFIGURATION_DSN		Data source name for Configuration database.
DIRECTORY_DSN		Data source name for Directory database.
OPERATIVE_DSN		Data source name for Operative database.
OUTBOUND_DSN		Data source name for Outbound database.
IP		IP address of virtual unit.
WebPort	80	Port of web server.
RDI	<a href="http://\$VU_IP:\$VU_WEB_PORT\$/RDI/Service.asmx">http://\$VU_IP:\$VU_WEB_PORT\$/RDI/Service.asmx</a>	URL of RDI web service. OII calls when SAPphone function <code>SPS_STAT_DATA_GET</code> is called.





## 4.2 OII Specific Registry Settings

First ones of the following registry settings are added during installation by the IA tool into the key HKEY\_LOCAL\_MACHINE\SOFTWARE\Wicom\VU1\OII. The latter optional part can be added manually in the registry.

Setting	Default value	Description
IpPort	21097	The port that OII uses with AS and Chat Server connection.
LogLevel	warning	The log level (info, trace, debug).
LogKeepLogDays	14	The number of days for saving old log files (before they are deleted).
UseSecureLogging	1	By default confidential information is not logged. Change to 0 to log all information.
<b>The remaining settings are optional:</b>		<b>These have to be edited into registry manually.</b>
MAP_SUBSCRIBE_URL		See the separate section about notification messages. There is now also database module parameter OII.MapSubscribeUrl which overrides this one.



ASYNC_SOAP	1	By default OII uses .NET asynchronous SOAP sending. Set to 0 to disable it.
SOAP_TIMEOUT_MS	100000	Milliseconds, the default is 100 seconds. How long OII waits for response from CRM before timing out. This is used in asynchronous SOAP messages in 7 SP8. The default of 100 seconds was used in synchronous SOAP messages already.
SOAP_WAIT_MS	250	Milliseconds. How long OII should wait before sending next SOAP event to same agent session, if the CRM has not responded to the earlier event yet.
SOAP_PROXY		The proxy server URL for SOAP events that OII is sending to CRM, e.g. <a href="http://proxyserver:80">http://proxyserver:80</a> . It is possible to configure SOAP proxy also in web.config.
MAIL_ROOT	C:\inetpub\mailroot	The directory used when sending e-mail messages. OII expects to find Pickup subfolder in here.  This setting is relevant only if database module parameter OII.UsePickupFolder is set to 1



		(enabled). By default MessageQueue table is used instead of Pickup subfolder for sending e-mails.
SERVICEPOINTMANAGER_EXPECT100CONTINUE	0 / False	The network options for notifications.
SERVICEPOINTMANAGER_USENAGLEALGORITHM	0 / False	The network options for notifications.
SERVICEPOINTMANAGER_MAXSERVICEPOINTIDLETIME	0	The network options for notifications.
ICIUSER_AUTOMATIC_UNSUBSCRIBE	1 / True	IciUser.getAttributes unsubscribes all old subscriptions.
ICIUSER_SETCURRENTQUEUES	0 / False	IciUser.setCurrentQueues is disabled by default so that the CDT queues are not set by SAP CRM. There is now also database module parameter OII.SetCurrentQueues which overrides this one.
ICIUSER_AUTOMATIC_DROP	1	Should OII drop existing calls when user logs into IC WebClient?



		0 = do not drop 1 = drop existing calls
--	--	--------------------------------------------

### 4.3 OII Module Parameters

OII module parameters are setup in IA tool for OII module in Integration Interfaces package installation. SC can display these parameters in System Management / Modules / OII. SAPphone parameters are setup in SC only.

System Management : Module : Integration Interfaces

Save Save and Close Reset Close

Back

Basic Information

Virtual Unit Name	VU7	Enable Action Channel Multi-Session	1
IP Address	10.31.96.65	Enable Outbound Integration	1
Type	Oii	Busy Status Duration (ms)	5000
Port	21097	Default Country Code	358
Enable OII Action Channel	1	Enable OII E-Mail Channel	1
Enable E-mail Channel Multi-Session	1	Enable Blind Transfer-and-End	1
Enable Receiving Presence Information via Chat	1	Enable Warm Transfer-and-End	1
Enable Sending DTMF Characters	1	International Dialing Prefix	00
SC_XFLD_Protocol	TCP	Enable Receiving Presence Information via E-Mail	1
Convert Incoming E-Mails	1	Enable OII Chat Channel	1
Ignore Outbound Calls Created in CDT	1	CAD Application ID	SAP_BCM/OII
Ignore Direct Agent Calls	1	Outbound Reject Option	SKIP
Show Busy Status in IC WebClient	1		
Wrap-Up after Failed Outbound Calls	1		
Prevent Not Ready Status with Calls and Chats	1		

SAPphone Configuration

Enable SAPphone Interface  
 Enable SAPphone Pop-Up with Incoming Call

Remove

Server Connection	Server Name	Client Connection	Ordinal ±
-aSAPphoneBCM -guscic7d -xsapgw20			1
-aSAPphoneBCM -guxciq0m -xsapgw76			2
			3

Some CEM parameters are configured in SC / System Management / Voice Channel.



System Management : Channels

Save | Reset

**Contact Center**

Queue Status Interval: 9 Seconds

Hide Queue Lists

Allow Queue Calls During Outbound Campaigns

**Voice Channel**

**Contact Management**

Number of Agents Affect Queue Length: 0.0

Allow Routing to Closed Overflow Queue

Define Entering Time: When Call Entered Contact C...

Enable Join Transfer: No

Show Original Source Number to External Agents

Timeout for Warm Transfer: 3600 Seconds

Enable Warm Transfer When Busy: No Warm Transfer if Target I...

Block Rejected Contact from Agent: Yes

Timeout for Join Transfer: 30 Seconds

Max. Waiting Time for Preferred Agent: 10 Seconds

**Reporting and Monitoring**

Max. Number of Allowed External Call Loops: 4

Timeout for Allowed External Call Loops: 60 Seconds

Timeout for Calls Connected to External Numbers: 30 Seconds

No. of Queue Calls Sent to and Displayed on Virtual Phone: 10

Delayed Acceptance Time: Milliseconds

Ringling Time for Agent's Direct Queue when Agent Busy: 20 Seconds

Indicate If Original External Source Number Is Unknown

**Addnl. Data Included When Allocating Calls**: XML,FirstBName,FirstBNumber

**Addnl. Data Included in Outbound Calls**: XML,FirstBName

Addnl. Data Included When Connecting Calls: 0

Addnl. Data Included When Supervising Calls: test

**Addnl. Data Included When Completing Consultation Calls**: XML,FirstBName

### 4.3.1 OII Module Parameters

Setting	Default value	Description
Integration Interfaces category:		
IpPort		The port number for the AS–OII communication. There is no default value, but a value 21097 is commonly used.



OII.Email	0	Defines whether the e-mail channel is enabled for all users. Set the value to 1 to enable e-mail channel.
OII.Chat	0	Defines whether the chat channel is enabled for all users. Set the value to 1 to enable chat channel.
OII.Action	0	Defines whether the action channel is enabled for all users. Set the value to 1 to enable action channel.
OII.IDD	00	International Direct Dialing prefix for phone numbers. This is removed from the incoming phone number and replaced with + character.  00 = Europe, default 011 = USA 001 = Asia  For example with the default value: 001234567 => +1234567
OII.Country	49	The default country code. If incoming phone number starts with 0 (single zero, not with OII.IDD, which is checked first), then the single zero is replaced with + and this default country code.  49 = Germany 358 = Finland



		<p>1 = USA</p> <p>...</p> <p>For example with the default value: 01234567 =&gt; +491234567</p>
OII.IgnoreOperDirect	1	<p>Defines if direct agent-to-agent calls be alerted in IC WebClient.</p> <p>By default, only the queue calls are offered to agents, and direct agent calls do not disturb customer processing. These direct calls can still be answered in CDT. If an agent forwards a customer call (that has originally come to a queue) to another agent, the call is alerted in IC WebClient as well.</p> <p>If 0, then all calls alert in IC WebClient.</p>
OII.CADApplicationID	SAP_BC M/OII	<p>Defines the Contact Attached Data application ID that OII uses for Contact Center data. If the information should be automatically transferred to SAP CRM system, use the value CRM_IC/BUPA.</p>
OII.CallRecordingCad	0	<p>Defines if call recording file name is added to CAD. If this is enabled, then there is a separate phoneCallChanged event containing CallRecording value. See Call Attached Data for more information.</p> <p>If 1, then CallRecording value is added to CAD.</p>



OII.Afterwork	1	<p>Defines if OII use Afterwork or Paperwork with phone calls and chats? Previously Paperwork was always used, now it is used only with E-mail and Action channel.</p> <p>If 0, then Paperwork is always used.</p>
OII.ShowPresenceProfiles	0	<p>Enable to show presence profiles in addition to absence profiles.</p>
OII.GetUserPresenceEmail	0	<p>CRM 7 has new function to do directory search and return presence information for users. First implementation supports only phone channel, while CRM 7 EhP1 supports also Chat channel (probably not E-mail channel).</p> <p>If 1, then E-mail channel is enabled.</p>
OII.GetUserPresenceChat	1	<p>CRM 7 has new function to do directory search and return presence information for users. First implementation supports only phone channel, while CRM 7 EhP1 supports also Chat channel as IM (instant messaging).</p> <p>If 0, then Chat channel is disabled.</p>
OII.ConvertEmail	0	<p>Because IC WebClient does not properly support embedded e-mails, OII converts these to regular attachments by default. OII also generates plain text body if there only is</p>





		<p>html body, and removes comma from e-mail addresses.</p> <p>If 0, then incoming e-mails are not converted.</p> <p>Default value changed to 0 in Cctr 7 FP13, as the HTML conversion can be slow with large e-mails, and latest CRM versions handle HTML correctly, so no need for the conversion.</p>
OII.UsePickupFolder	0	<p>By default MessageQueue table is used for sending e-mails.</p> <p>If 1, then IIS Pickup folder is used instead.</p>
OII.ShowBusyStatus	0	<p>When outbound call fails because of Busy status, OII can send Busy status instead of the normal Ended. Because there is no way to clear the Busy call from IC WebClient, OII also sends Ended status after 5 seconds. This behavior is disabled by default.</p> <p>If 1, then Busy status is shown in IC WebClient.</p>
OII.BusyStatusDuration	5000	<p>Defines how long the Busy status is shown in IC WebClient, in milliseconds. The default is 5 seconds.</p>
OII.IgnoreCdtOutbound	1	<p>By default outbound calls created in CDT are not shown in IC WebClient.</p>



		If 0, then these calls are shown in IC WebClient.
OII.SendDTMF	1	SendDTMF is supported in CRM 2007 (CRM 6.0) and later, but not in CRM 5.  If 0, then SendDTMF capability is not used. Set to 0 when CRM 5 is used.
OII.RequestedWrapUp	0	By default the automatic wrap up mode is used, meaning that when connected call disconnects, the call goes to Wrap Up processing status.  If 1, then requested wrap up mode is used, so that call goes to Wrap Up processing status only if client calls wrapUpRequired.
OII.WrapUpEndedFromCdt	0	By default OII does not end Wrap Up states for contacts (phone calls, chats) when CDT goes to Ready.  0 = Don't end Wrap Up when CDT goes to Ready. 1 = End Wrap Up when CDT goes to Ready.
OII.WrapUpOutbound	0	By default only Connected outbound calls go to Wrap Up state.  If 1, then all outbound calls (even failed) go to Wrap Up state.



OII.PhoneMultiSession	0	<p>Define if an agent can receive additional calls when already processing a phone call (in Wrap Up).</p> <p>If 1, then multiple calls can be processed at same time.</p>
OII.ActionMultiSession	0	<p>By default OII keeps CDT in Paperwork when Action Item is being processed.</p> <p>If 1, then OII changes CDT back to Serving when Action Item is accepted, so that agent can process simultaneous phone calls and chats.</p>
OII.EmailMultiSession	0	<p>By default OII keeps CDT in Paperwork when E-mail (message) is being processed.</p> <p>If 1, then OII changes CDT back to Serving when E-mail (message) is accepted, so that agent can process simultaneous phone calls and chats.</p>
OII.PdBeginPreview	0	<p>By default OII does not send Outbound Preview state call to CRM, because CRM supports this only in CRM 7.0 EhP1.</p> <p>If 1, then OII sends Preview status (111) to CRM when Contact Center Outbound call is initiated in preview state. In order for this to work, the campaign should be imported first from CRM call list, using SAPphone function SPS_PDLISTTRANSFER. The ICI call will</p>



		<p>have attached data with application id “CRM_IC_CLM_PVDIAL”. There will be single parameter “CALLID”, which has the value of the call list entry from CRM.</p>
OII.PdEndPreviewReject	SKIP	<p>By default OII uses SKIP option when outbound preview call is rejected in IC WebClient. This means that the customer is just skipped for now, but called again on next round. Call result is SKIP.</p> <p>If REJECT, then the customer is not called anymore. Call and customer result will be AGENT_REJECTED.</p>
OII.PdRescheduleImmediately	0	<p>By default OII does not end the outbound call when CRM sends rescheduling information in CAD, if the call is connected or in wrap up.</p> <p>0 = Call is ended only in Preview state. 1 = Call is ended also in Connected / Wrap Up states.</p>
OII.ForwardAndEnd	0	<p>By default OII wraps up connected calls (Wrap Up / Ended), but if this is set to 1, then call is directly ended (Not In Process / Ended) when Forward (Blind Transfer) is called. This is to fix IC WebClient problem in Transfer &amp; End button (it does not do End for OII).</p>
OII.TransferAndEnd	0	<p>By default OII wraps up connected calls (Wrap Up / Ended), but if this is set to 1, then call is directly ended (Not In Process / Ended)</p>



		when Transfer (Warm Transfer) is called. This is to fix IC WebClient problem in Warm Transfer & End button (it does not do End for OII).
OII.GetPresenceQueueInfoAll	0	By default OII returns only user's queues in IciSystem.getPresenceQueueInfo function.  0 = return only user's queues 1 = return all queues in system
OII.GetPresenceQueueInfoFilter	0	By default the returned queues are not filtered in IciSystem.getPresenceQueueInfo function.  0 = no filter 1 = return only queues that have logged in agents 2 = return only queues that have ready agents
OII.LoginDropCalls	1	Should OII drop existing calls when user logs into IC WebClient?  0 = do not drop 1 = drop existing calls
OII.SetCurrentQueues	0	IciUser.setCurrentQueues is disabled by default so that the CDT queues are not set by SAP CRM. This value overrides the registry value ICIUSER_SETCURRENTQUEUES. Older CRM versions set all queues to active when IC WebClient is started, and all to disabled when IC WebClient is closed. The



		<p>newer CRM versions don't do this anymore, instead there is a possibility to set the queues manually.</p> <p>0 = do not set queues 1 = set queues from CRM</p>
OII.MapSubscribeUrl		<p>See the separate section about notification messages. This value overrides the registry value MAP_SUBSCRIBE_URL.</p>
OII.ReverseDnsSubscribeUrl	0	<p>This parameter is for changing the IP address to host name in URL given by CRM in Subscribe command.</p> <p>If 1, then IP address is changed to host name using Reverse DNS Lookup.</p>
OII.SetExtraData	1	<p>If disabled (value = 0), CAD is sent to CEM only when call is transferred. If enabled (value = 1), the CAD is sent to CEM immediately when CRM sets it, so that the phone call can be closed by agent and CEM still gets the CAD for Survey IVR.</p>
OII.CheckDuplicateActionItem	0	<p>If disabled (value = 0), then OII does not check for duplicate action items when CRM is routing them. If enabled (value = 1), then OII checks if the processId matches earlier action item and does not create a new action item in this case.</p>



		<p>This is improved in version 7SP10 in the following way:</p> <ol style="list-style-type: none"> <li>1. If this option is enabled, then the action item duplicate is searched from database as well, if it does not exist in memory.</li> <li>2. If the action item contains an attribute DELETE = X, then the action item is not created but routing is ignored. If this option is enabled, and if the duplicate action item exists, then it is marked as Deleted.</li> <li>3. If a user is currently handling the action item that is being deleted in CRM, then the user is notified that the action item is now Ended.</li> </ol>
The following parameters affect the optional OII SAPphone interface		Added in 6.0SP5. See <a href="#">Configuring IC on SAP CRM</a>
OII.SAPphoneEnabled	0	<p>Defines whether SAPphone interface (both server and client) is enabled in OII.</p> <p>If 1, then SAPphone interface is enabled. Connection strings (both server and client) must be configured with separate parameters, see below.</p>
OII.SAPphonePopup	1	Defines whether there will be a SAPphone UI popup in incoming call.



		<p>If 0, then there is no popup.</p>
<p>OII.SAPphoneServer</p>		<p>Defines the SAPphone server connection string in the format:</p> <p><i>-a[Program ID] -g[Gateway Host] -x[Gateway Service]</i></p> <p>where [Program ID], [Gateway Host] and [Gateway Service] (=sapgw and the two-digit service number) are the settings defined in the CRM RFC configuration.</p> <p>For example:</p> <p><code>-aBCM_VU1 -g1.2.3.4 -xSAPGW20</code></p> <p>Program id should be unique, if there are more than one Contact Center systems connecting to the same CRM/ERP system.</p> <p>It is possible to define more than one server connection, by adding OII.SAPphoneServer2, OII.SAPphoneServer3 etc. In Contact Center 7 SP4, there is a new connection string option REG_COUNT to make more than one server registration. For example, the following creates 20 connections:</p> <p><code>-aSAPphoneBCM -g1.2.3.4 -xsapgw76 REG_COUNT=20</code></p> <p>To use new SAP .NET Connector 3.0, give the connection string in the following format:</p>





		<p>PROGRAM_ID=SAPphoneBCM</p> <p>GWHOHOST=uxciq0m GWSErv=sapgw76</p>
OII.SAPphoneServerName		<p>Required for OII client functions and created automatically when CRM/ERP calls XCHGPARAMS server function (SERVER_NAME parameter). Can be edited or removed manually.</p>
OII.SAPphoneClient		<p>Defines the client connection string, in the format CLIENT=[client] USER=[uid] PASSWD=[pwd] LANG=EN ASHOST=[Gateway Host] SYSNR=[service number] where all parts in square brackets must be same as defined in the CRM user and RFC settings.</p> <p>It is possible to define more than one client connection, by adding OII.SAPphoneClient2, OII.SAPphoneClient3 and so on.</p> <p>For example:</p> <p>CLIENT=506 USER=bcm1 PASSWD=bcm1 LANG=EN ASHOST=1.2.3.4 SYSNR=20</p>
Category: CEM		<p>OII adds these default values to configuration database when it is started for the first time. It is possible to define some of these parameters also at queue level.</p>
ExtraDataWhenAllocated	XML,FirstBName,FirstBNum	<p>Defines whether attached data is used in new calls.</p> <p><b>NOTE:</b> In Siebel integration, this value should</p>



	er,SIP_C ALL_ID	be XML,FirstANumber,FirstBName,FirstBNumbe r  (SAP integration does not need FirstANumber).
ExtraDataWhenCallOut	XML,First BName,S IP_CALL _ID	Defines whether the Attached Data information is stored in call (XML). Attached Data contains information about the customer in IC WebClient that agent has selected for the call, when the call is transferred to another agent.
ExtraDataWhenConnecte d	SIP_CAL L_ID	Extra data when call is connected.
ExtraDataWhenJoin	XML,First BName,S IP_CALL _ID	Extra data when warm transfer is completed.

### 4.3.2 Queue Parameters

These parameters can be set in SC > Queue Management.

Setting	Default value	Description
---------	---------------	-------------



AnswerInfo		The template for the informative pop-up window related to an inbound call or e-mail. See the separate section about queues.
OII	1	Defines whether the integration is enabled (1) or disabled (0) for the queue.

### 4.3.3 Pseudo Queue

There can be a pseudo queues with the technical name OII\_EMAIL. If the queue exists in the database, then only those users that have this queue activated receive notification messages for new e-mails. If the OII\_EMAIL queue does not exist, then all users receive new e-mail notifications (if the e-mail channel is enabled in other settings, see OII.Email application parameter).

Normally users can activate or deactivate queues in the CDT application on the run. If the OII\_EMAIL queue is not activated for the user, then the user can process e-mails by using the CDT application.

There can also be similar pseudo queues for chat and action channel: OII\_CHAT, OII\_ACTION.

## 4.4 Notification Messages

The interface sends notification messages about changed users, phone calls, and e-mails back to SAP CRM (after it has subscribed to these). It gets the endpoint address in the subscriber message, and it does not need to be the same as the client endpoint that calls the subscribe function.



### 4.4.1 OII.MapSubscribeUrl Parameter

The subscribe endpoint may not work for some reason, for example, if the IP address is internal to the customer. You can define the mapping with the following module parameter in IA:

OII.MapSubscribeUrl => The value should be <subscribed IP address and port, real IP address and port>, for example:

```
127.0.0.1:80, server1.com:80
```

There can be several value pairs separated by , (comma), for example:

```
127.0.0.1:80, server1.com:80, 127.0.0.1:50001, server2.com:80
```

It is possible to give this parameter also in registry (MAP\_SUBSCRIBE\_URL), in which case OII.MapSubscribeUrl overrides the registry value.

### 4.4.2 OII.ReverseDnsSubscribeUrl Parameter

Contact Center 7 SP4 has new OII.ReverseDnsSubscribeUrl parameter for changing the IP address to host name in URL given by CRM in Subscribe command.

The problem with IP address is that if secure HTTPS is used, the certificate is usually created for the host name, so IP address won't work in URL. Still CRM only gives IP address.

The default value is:

```
OII.ReverseDnsSubscribeUrl = 0
```

Meaning that reverse DNS lookup is not done.

OII.ReverseDnsSubscribeUrl is alternative to OII.MapSubscribeUrl, which can be used to do any modification to URL string (also port number can be changed).



## 4.5 SOAP Proxy

Another option is to use SOAP\_PROXY to send SOAP events to proxy server, which then sends them to CRM. The registry setting SOAP\_PROXY can also be configured in web.config.

The following example sets the default proxy server address to `http://proxyserver`, indicates that the proxy should not be used for local addresses, and specifies that all requests to servers located in the `contoso.com` domain should bypass the proxy.

```
<configuration>
  <system.net>
    <defaultProxy>
      <proxy
        usesystemdefault = "false"
        proxyaddress = "http://proxyserver:80"
        bypassonlocal = "true"
      />
      <bypasslist>
        <add address="http://\[a-z\]+\contoso.com/" />
      </bypasslist>
    </defaultProxy>
  </system.net>
</configuration>
```

## 4.6 Users

The integration requires that the user has the same user ID (login name, or alternatively CRM ID) both in the Contact Center and SAP CRM systems. E-mail and chat channels require that an e-mail and chat address has been defined for the user in SC. If a user can start CDT, it is basically ready for OII. IM chat (Instant Messaging) requires that the chat address must be identical to e-mail address (CRM limitation).



By default, the phone channel is enabled and e-mail, chat and action channels are disabled for all users. E-mail channel can be enabled for all users with the application parameter OII.Email=1.

This global e-mail channel setting can be overridden (enabled or disabled) for each user by using the OII\_EMAIL pseudo queue, as described above.

## 4.7 Queues

### 4.7.1 Enabling and Disabling

By default all queues are enabled, but you can disable a queue with the queue parameter OII (Value = 0). If the parameter value is changed, the Apply must be clicked on the queue level. All database settings are reloaded each midnight as well.

### 4.7.2 Informative Pop-Up Window

You can define that an informative pop-up window appears when there is an inbound queue call. With OII the pop-up window is supported for inbound e-mail messages as well.

Enable and define the pop-up window by using the AnswerInfo queue parameter in the Communication Desktop (CDT) category.

Enter the message text you want to display. The text can contain HTML, and a value for an e-mail queue can contain a special value \* that means that OII uses the following template in pop-up window:

```
<TABLE id=email cellSpacing=1 cellPadding=1 border=0 STYLE="font-size:8pt">
<TR><TD>Queue:</TD><TD><b>{QUEUE}</b></TD></TR>
<TR><TD>From:</TD><TD>{FROM}</TD></TR>
<TR><TD>To:</TD><TD>{TO}</TD></TR>
<TR><TD>Subject:</TD><TD>{SUBJECT}</TD></TR>
<TR><TD colspan=2><HR>{BODY}</TD></TR>
</TABLE>
```



The {QUEUE}, {FROM}, {TO}, {SUBJECT}, and {BODY} tags can only be used with e-mail queues.

### 4.7.3 Action Item Queue

By default all Action Items are queued into the ACTION queue. This queue is created by OII automatically. Display name is “Action Queue” and it has the following Address: ACTION

Queue Management : E-Mail Queue : Action Queue							
Basics							
Name	Action Queue			Queue Mode	Auto-Allocation		
Description	The default Action channel queue used in CRM integration			Warning Time	Seconds		
Ordinal	2			ID	001EAA7D6A1E4E4889FC03F17F649E3C		
Prompt Language				Script	Open		
Time Zone				Opening Type			
Critical Time	Seconds			<input type="checkbox"/> Directory Synchronization			
Numbers/Addresses							
Delete							
Address	Priority	Extension	Language	E-Mail Account	E-Mail Password	Sender	E-Mail Server
ACTION						<input checked="" type="checkbox"/>	

It is possible to direct incoming ERMS Push Action Items to other e-mail queues, too. If Action Item contains information about the original TO e-mail address (in itemData/actionItemAttributes, where id = “TO\_ADDRESS”), then this address is used to find the correct e-mail queue. It is possible to add this address to any queue, by adding the following channel (if the address is [erms@company.com](mailto:erms@company.com), for example):

Queue type: E-Mail

Number/Address: [ERMS@COMPANY.COM](mailto:ERMS@COMPANY.COM)

E-mail server: NONE



## 4.8 Automatic Detection of System Configuration Change

OII monitors ChangeTrack tables and automatically reloads user or queue if it has been changed in database. OII also detects when OII parameters have been changed in IA.

In addition to that, OII also reloads all settings each midnight (local server time).





# 5 Configuring IC on SAP CRM

## 5.1 Create RFC Destination

In transaction SM59 create a new HTTP RFC connection to your Contact Center system (192.168.12.15 in our case).

Remember the name of the Connection since you need to provide it as input later on.

### RFC Destination BCM

Connection Test

RFC Destination	BCM	
Connection Type	G HTTP Connection to External Serv	Description

**Description**

Description 1	RFC Connection to OII interface SAP BCM
Description 2	
Description 3	

Administration
Technical Settings
Logon & Security
Special Options

**Target System Settings**

Target Host	192.168.12.65	Service No.	80
Path Prefix	/OII/OII.asmx		

**HTTP Proxy Options**

Global Configuration

Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

Test the connection.



## Connection Test HTTP Destination BCM

Detail	Value
Status HTTP Response	200
Status Text	OK
Duration Test Call	269 ms

The G type RFC could be configured with active SSL, thus HTTPS would be enabled in OII for secured connection. This involves uploading the OII related SSL certificate to CRM system, and configuring the Contact Center web server to use HTTPS. Thus, the values in 'Connection' and 'Session Connection' of 'Maintain Communication Management Software Connections' transaction CRMM\_BCB\_ADM should point to the G type RFC with active SSL & the H type RFC destination pointing to the CRM application server and HTTPS port respectively.

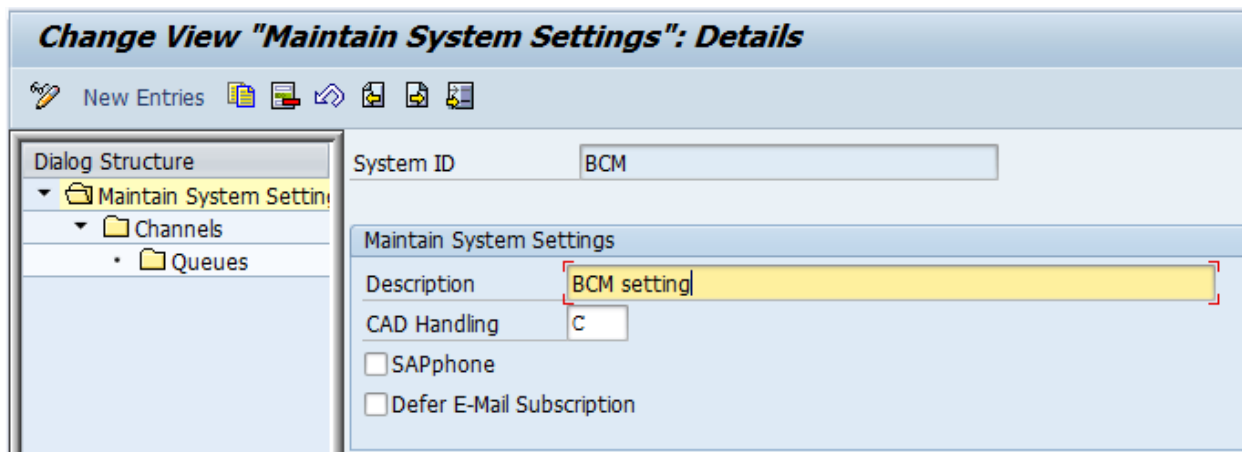
## 5.2 Define Contact Center as a Communication System

Add Contact Center as a Communication Management System via Transaction CRMM\_IC\_MCM\_CCADM.

Add a Channel for the created entry.

CAD Handling should contain value "C" (Inherited / Copied Back) (empty means Not Inherited / Not Copied Back). This setting affects the interaction record handling when a contact is transferred from agent to another, ideally the Interaction Record (IR) transfers while call transfers. If it is set to "C" then the CAD is set for the original call. When the transfer is completed, then the CAD is set for the consultation call.

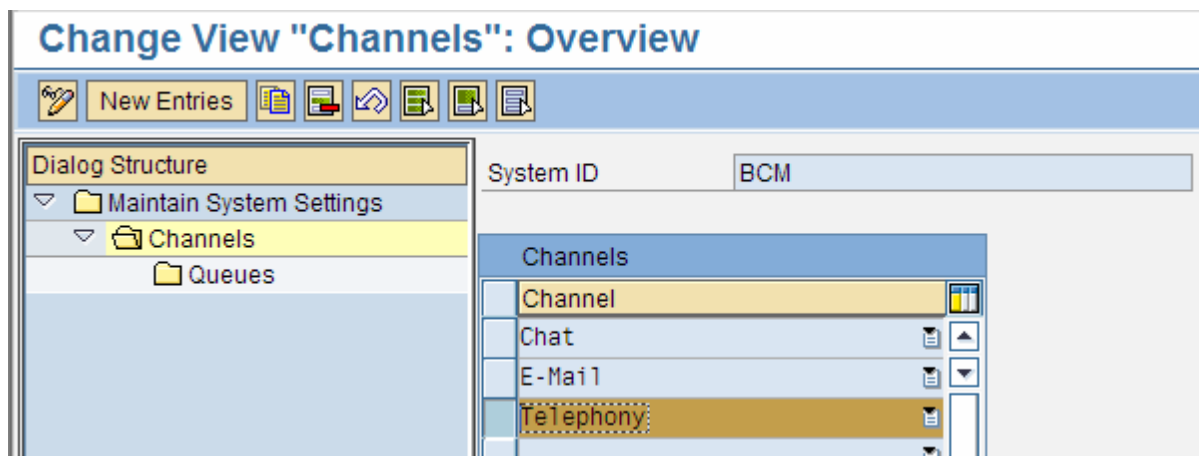
The result should like look the picture below.



Highlight the entry on the right and click on Channels.

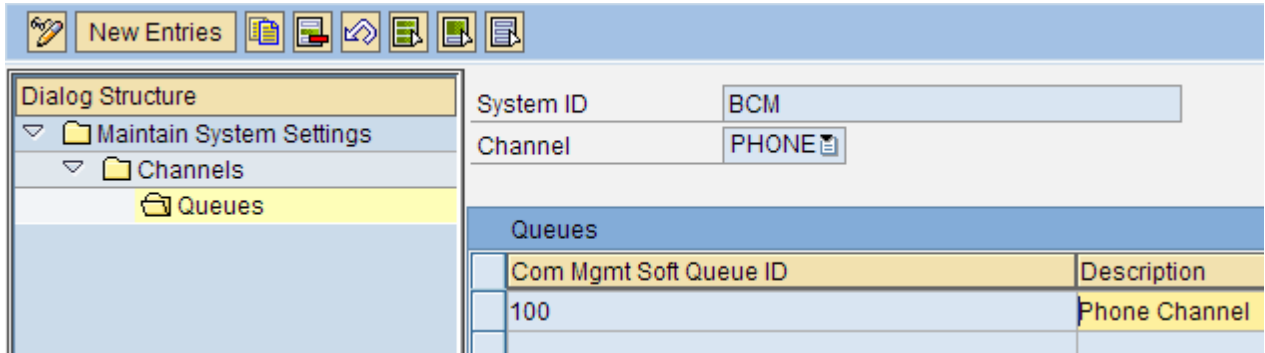
Add a new telephony channel (E-mail and chat are optional).

The result should like look the picture on the next page.



Highlight the Telephony Channel and click on Queues to add a queue.

The result should like look the picture below. Queue ID must be the address of queue, so phone number in case of phone queue.

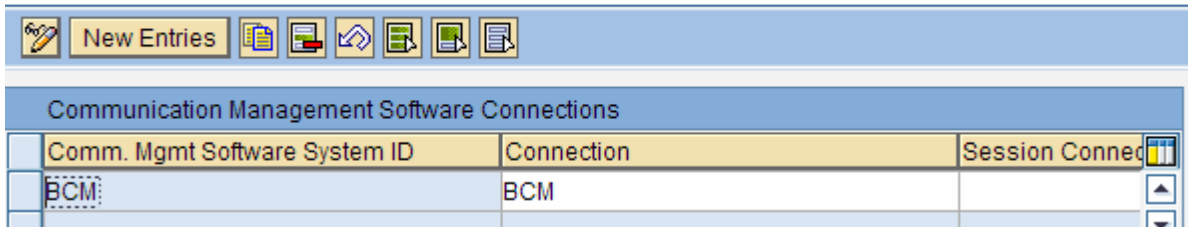


### 5.3 Define Connections

Start Transaction CRMM\_BCB\_ADM to link the RFC connection to the Communication Management Software by creating a new entry.

The result should like look the picture below.

#### Change View "Communication Management Software Connectic



### 5.4 Define Profiles

Start transaction CRMC\_IC\_MCM\_CCPRO and create a New Entry.

Checkboxes could be checked-in if the customer needs presence & multisession features.

The result should like look the picture below.

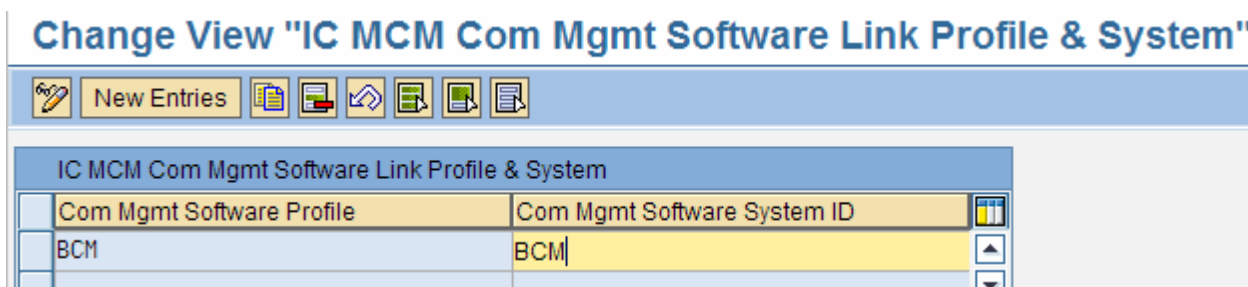


Define Communication Management Software Profiles				
Com Mgmt Software Profile	Description	Man Logon	Presence	Multi-Session
BCM	BCM CMS PROFILE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## 5.5 Assign System to Profile

Start Transaction CRMM\_IC\_MCM\_CCLNK and create a New Entry.

The result should like look the picture below.



## 5.6 Business Role Configuration

Start transaction CRMC\_UI\_PROFILE (or go to IMG by transaction SPRO, then click on button SAP Reference IMG).

Navigate to the Business Role to be configured for Contact Center integration, this role could either be a new Business role or a copy of an existing Business role (ex: IC\_AGENT role) with Role Config Key, Nav Bar Profile, Layout Profile, Technical Profile and PFCG Role ID configured as per requirement.

Select the role & click on Assign Function Profiles in the left window pane.

Set the CONTACTCENTER parameter from available values, this is based on entry made in transaction CRMC\_IC\_MCM\_CCPRO and save the profile.



## Change View "Assign Function Profiles": Overview

**Dialog Structure**  
 Define Business Roles  
     Assign Function Profiles  
     Adjust Work Centers  
     Adjust Work Center Group Links  
     Adjust Direct Link Groups  
         Adjust Direct Links  
     Define Keyboard Shortcuts  
     Adjust Central Search Objects

Business Role:

Assign Function Profiles	
Function Profile ID	Profile Value
BPIDENT	YBP_B2B
CAD	
CALL_LIST	AUTO_PARTNER_CONFIRMATION
CAMPAIGN	
CASE	DEFAULT
CATEGORIZATION	DEFAULT
CONTACTCENTER	BCM
CONTENT MGMT	DEFAULT

## 5.7 Business Role Customizations

In addition to the feature specific customizations, the entries related to Business role needs to be customized so the features like Improved Context Transfer, Instant Messaging, Preview dialing work fine in the CRM IC for the specific Business role to be used with Contact Center integration.

## 5.8 Set Profile Parameter for User

Start transaction SU01

Open the user for editing to which you want to assign the profile.

Add the parameter CRM\_UI\_PROFILE with the name of the role you just created as a value.



## Maintain User

User	SBLEIJ			
Last Changed On	BEHEER	21.08.2008	14:44:23	S
<div style="display: flex; justify-content: space-around;"> <span>Address</span> <span>Logon data</span> <span>SNC</span> <span>Defaults</span> <span>Parameters</span> </div>				
Parameter				
	Parameter ID	Parameter value	Short Description	
	CRM_UI_PROFILE	BP_ICA_BCM	CRM UI Profile	

It is also possible to relate the 'Business role' to CRM 'User' through a 'Position'. The CRM 'User' needs to be assigned to a 'Position' using transaction 'PPOMW' (ideally the 'Position' belongs to an 'Organizational unit' in CRM system), this 'Position' is linked to the 'Business Role'.



# 6 Configuring SAPphone on SAP CRM/ERP

SAPphone is an optional interface that OII offers for enabling basic telephony calls in the CRM/ERP user interface.

## 6.1 RFC Destination

Define the RFC destination of type T:





## RFC Destination BCM\_SAPPHONE\_HARRI

Connection Test
Unicode Test

<b>RFC Destination</b>	BCM_SAPPHONE_HARRI	
<b>Connection Type</b>	T TCP/IP Connection	Description

**Description**

Description 1	BCM SAPphone Harri Pesonen
Description 2	
Description 3	

Administration    Technical Settings    Logon & Security    MDMP & Unicode    Special Options

**Activation Type**

Start on Application Server                       Registered Server Program  
 Start on Explicit Host  
 Start on Front-End Work Station

**Registered Server Program**

Program ID	SAPphoneBCM
------------	-------------

**Start Type of External Program**

Default Gateway Value  
 Remote Execution  
 Remote Shell  
 Secure Shell

**CPI-C Timeout**

Default Gateway Value  
 Specify Timeout                       Defined Value in Seconds

**Gateway Options**

Gateway Host	<input style="width: 90%;" type="text" value="uscq7d"/>	<span style="border: 1px solid #ccc; padding: 2px 5px; margin-left: 10px;">Delete</span>
Gateway service	<input style="width: 90%;" type="text" value="sapgw20"/>	

For the above destination, the correct SAPphone server connection string in the Contact Center parameter would be:

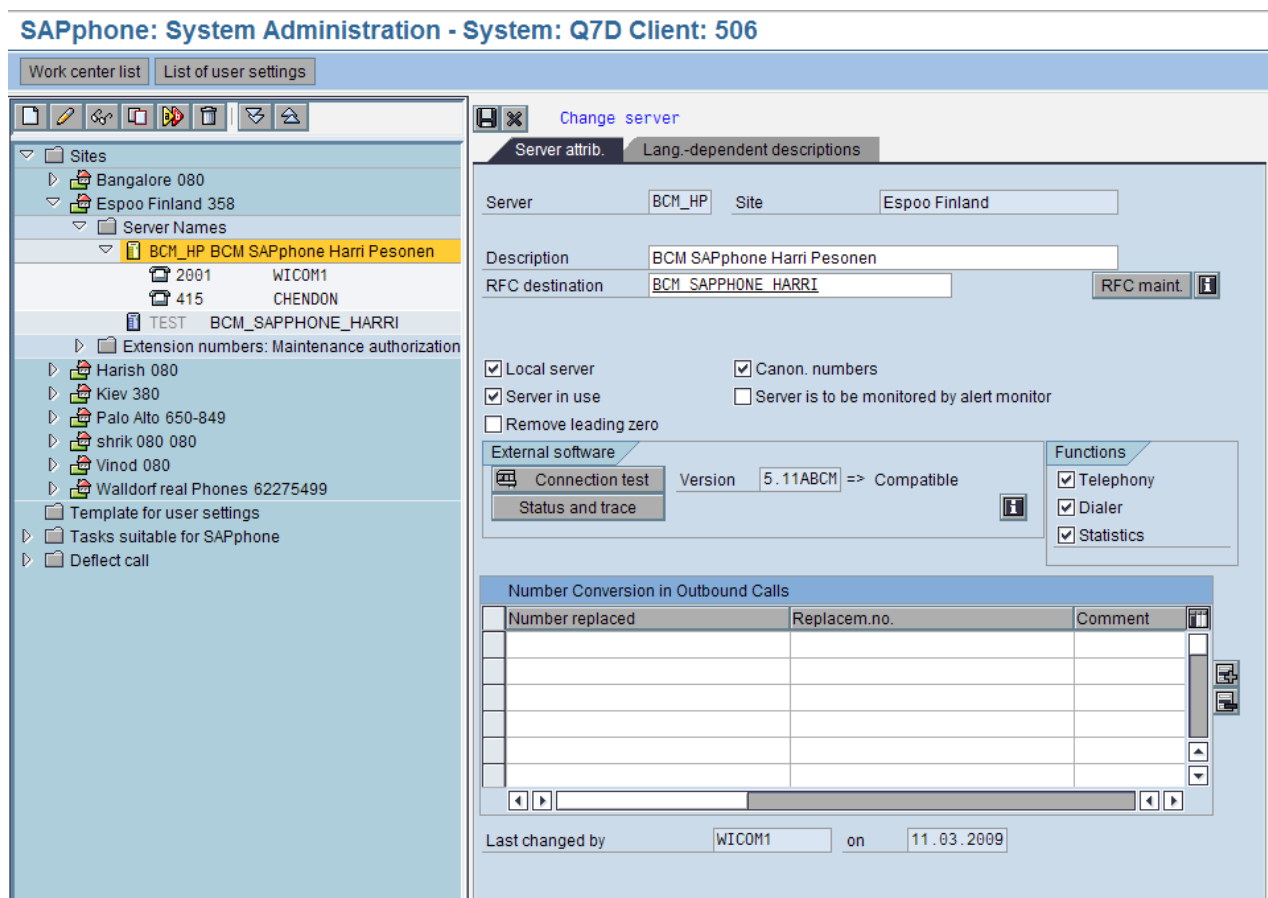
OII.SAPphoneServer = -aSAPphoneBCM -guscq7d -xsapgw20



## 6.2 SAPphone Server

Transaction SPHB starts SAPphone System Administration UI.

On contrary to the picture below, the checkbox ‘Local Server’ needs to be unchecked. If it’s checked in, the SAPphone automatic popup will not show up during incoming phone call (even if other setting related to the popup is in place).



## 6.3 SAPphone Administration

Transaction SPHA starts SAPphone Administration.



## SAPphone: Administration

Work center    User settings

Initiate outbound call

Simulate inbound call

SAPphone

## 6.4 Work Center Settings

### SAPphone: Work Center-Specific Settings

Copy settings    Connection test

Work center-specific settings

Work center: 102    Ty.: User-dependent

1. Telephone no.: 2001     For ACD functions     Default

2. Telephone no.:     For ACD functions     Default

Additional tel.numbers

Telephony server: [BCM\_HP] BCM SAPphone Harri Pesonen

User responsible: WICOM1

Current User: WICOM1

Host name:



## 6.4.1 User-Specific Settings

### SAPphone: User-Specific Settings

Available tasks (incoming and outgoing)

User

**Display Inbound Calls**

No Display

Display Inbound Call

Start task immediately

Task

Only start task if caller can be determined

**Notification of call in absence**

Message if absent for call

No expiration time     Expire After  Days

**Data Areas for Caller Search in Inbound Calls**

Pri...	Data Area	

**Number group for forwarding unanswered calls**

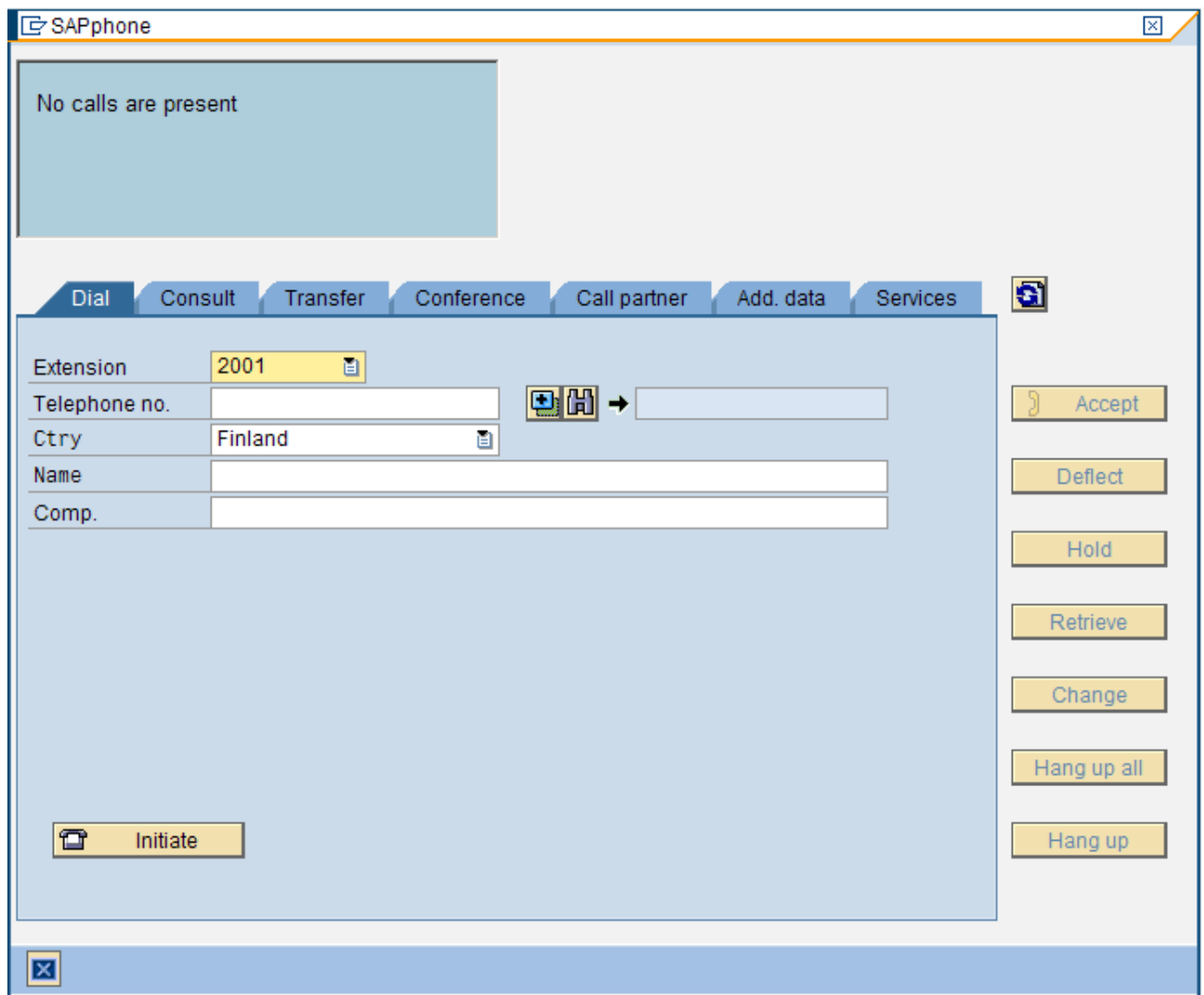
Group

**Call center logon data**

User  Password  /



## 6.4.2 SAPphone UI





# 7 ICI Functionality Overview

The interface is meant for SAP CRM ICI (*Integrated Communication Interface*). It implements the ICI interface which is used with SAP CRM IC (*Interaction Center*) Web Client. The OII interface is compatible with the ICI specification version 3.07 (29.09.2010), but it does not implement all ICI features. OII also contains some functions that are not documented in the latest ICI specification.

## 7.1 ICI Interfaces

The following ICI interfaces are supported:

Interface	Functions	Notes
IciSystem	exchangeProductInformation  getWorkcenterCapability  getPresenceQueueInfo  getUserPresence	“exchangeProductInformation “ returns OII version. This is called by SAP CRM 4.0 when OII URL is configured in CRM. Not used in CRM 5.0 or later.  “getWorkcenterCapability” returns always empty result.  “getPresenceQueueInfo” is new in Contact Center 6.0.  “getUserPresence” is new in Contact Center 6.0 SP1, used by CRM 7.0.
IciUser	getAttributes setWorkmode	When a user logs into SAP CRM, first ICI calls



	<p>setCurrentChannels                  setCurrentQueues                  setAddresses                  subscribe                  unsubscribe</p>	<p>getAttributes and then these other functions.</p>
IciEvent	<p>userChanged                  subscriptionEnded                  containerChanged                  phoneCallChanged                  messageChanged                  chatChanged                  chatNewPosting                  actionItemChanged                  monitorDataChanged</p>	<p>After subscribe, OII sends these notification messages when a user/call/message/container changes.</p> <p>This is the only client-side interface; all other interfaces are services.</p>
IciItem	<p>create                  wrapUpRequired                  wrapUpEnded                  accept                  reroute                  forward                  getAttachedData                  setAttachedData                  deleteAttachedData</p>	<p>An item can be either a phone call or an e-mail message.</p>
IciContainer	<p>subscribe                  unsubscribe</p>	<p>A container can be either a phone line or an e-mail folder.</p>



IciPhoneLine	getCalls dropCalls	Returns all current phone calls in a certain line.
IciPhoneCall	getAttributes dial consult transfer reconnect drop hold unhold alternate conference dropParty	Not implemented: listen, and listenEnd.
IciFolder	getMessages	Gets the list of e-mail messages that are in one folder.
IciMessage	getAttributes getContent setContent delete send setActionFlags end	Functions for e-mail channel messages. “end” is used with Action channel, new in Contact Center 6.0.
IciChatLine	getSessions	Returns all active chat sessions for user.





IciChatSession	getAttributes postMessage getDialog invite leave	ICI 3.07 specification has also: join, conference. These are currently not implemented because it is not possible to call these in IC WebClient.
IciActionItem	route	Route new action item for Contact Center allocation. ICI 3.07 specification is out of date with Action channel.
IciMonitor	subscribe getInstances getCatalog unsubscribe	IciMonitor interface in OII 7.0 is not ready for production use. Monitor interface is not documented yet.

Each of these interfaces is implemented in separate web service, but because SAP CRM configures only one endpoint, OII rewrites the path internally. For example, in CRM the OII endpoint is defined as <http://10.31.100.50:1080/OII/OII.aspx> in the following screenshot:



## RFC Destination BCM\_TEST@ESPOO

Connection Test

RFC Destination	BCM_TEST@ESPOO	
Connection Type	G HTTP Connection to External Serv	Description

**Description**

Description 1	BCM Test at Espoo	
Description 2		
Description 3	Q7D to BCM Test servers at Espoo	

Administration
Technical Settings
Logon & Security
Special Options

**Target System Settings**

Target Host	10.31.100.50	Service No.	1080
Path Prefix	/OII/OII.asmx		

OII gets the SOAPAction header from each SOAP call, and changes OII.asmx according to the following table:

SOAPAction	Web service
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciSystemBean"	IciSystemService.asmx



"http://inqmy.com/soapdispatcher/rpc/bcbici/IciUserBean"	IciUserService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciContainerBean"	IciContainerService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciItemBean"	IciItemService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciPhoneLineBean"	IciPhoneLineService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciPhoneCallBean"	IciPhoneCallService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciFolderBean"	IciFolderService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciMessageBean"	IciMessageService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciChatLineBean"	IciChatLineService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciChatSessionBean"	IciChatSessionService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciActionItemBean"	IciActionItemService.asmx
"http://inqmy.com/soapdispatcher/rpc/bcbici/IciMonitorBean"	IciMonitorService.asmx



## 7.2 Item Attached Data

### 7.2.1 Call Attached Data (CAD)

OII automatically generates CAD for all incoming phone calls, according to the ICI specification. OII uses “SAP\_BCM/OII” to identify itself in XML. For example:

```
<ItemAttachedData>
  <Application id="SAP_BCM/OII">
    <FirstBNumber>101</FirstBNumber>
    <FirstBName>Queue 1</FirstBName>
    <BNumberName>Queue 1</BNumberName>
    <OrigQueue>Queue 1</OrigQueue>
  </Application>
</ItemAttachedData>
```

In Contact Center 7 queue names are replaced by GUIDs, and there is optional CallRecording value:

```
<ItemAttachedData>
  <Application id="SAP_BCM/OII">
    <FirstBName>C2DE7EC4-7886-466D-8A04-
1120845D5E57</FirstBName>
    <FirstBNumber>600</FirstBNumber>
    <BNumberName>C2DE7EC4-7886-466D-8A04-
1120845D5E57</BNumberName>
    <OrigQueue>C2DE7EC4-7886-466D-8A04-
1120845D5E57</OrigQueue>

    <CallRecording>2012_04\02_10_29_42_SSR_Cr_F5F047CF7DAF4AE1AA6E
556BF0D84A41.wav</CallRecording>
  </Application>
</ItemAttachedData>
```



The information that is added to CAD by OII is defined in Administrator Application parameter:

ExtraDataWhenAllocated = XML,FirstBName,FirstBNumber

XML is a special key that is needed by OII, it will contain the CAD in ICI XML format. Other keys defined in “ExtraDataWhenAllocated” parameter will be added to CAD automatically. Some keys are available in CEM automatically and others can be added by IVR.

Key	The first ones are added to CAD automatically.
FirstBNumber	Internal phone number of the queue that was first called.
FirstBName	Display name of the queue that was first called.
BNumberName	Technical name of the current queue.
OrigQueue	Technical name of the previous queue.
Key	The remaining is not added by default, unless they are listed in ExtraDataWhenAllocated parameter.
GUID	Call GUID. Value is the same as callHandle element in phoneCallChanged event, except that callHandle is without - characters (callHandle is 32 characters, GUID is 36 characters).
CALL_ID	Unique incrementing Id for the call, starting from 1 when CEM is started, like “CI_123L”.



FirstANumber	The original caller (customer) phone number.
IVRInfo	WACP encoded dictionary of data added by IVR.

See also OII.CADApplicationID parameter.

Call recording file name is added if parameter OII.CallRecordingCad is set. The CAD containing CallRecording value comes in separate phoneCallChanged event, so OII.CallRecordingCad parameter should be set to 0 (default) if value is not used.

### 7.2.2 SIP Call ID

SIP\_CALL\_ID was added to CAD in Contact Center 7.0 SP4. This requires that the following data is configured in SC / System Management / Channels / Voice Channel / Extra Data Settings:

Name	Value
ExtraDataWhenAllocated	XML,FirstBName,FirstBNumber,SIP_CALL_ID
ExtraDataWhenCallOut	XML,FirstBName
ExtraDataWhenConnected	SIP_CALL_ID
ExtraDataWhenJoin	XML,FirstBName,SIP_CALL_ID

### 7.2.3 Chat Attached Data

Similar to CAD, also chat channel has attached data, which is originally set by Internet Chat Client, which calls Chat Portal, which offers a similar ICI interface as OII. Chat Portal connects to Chat Server, which then sends chat events to OII and CDT. The format for chat attached data is:

```
<?xml version="1.0" encoding="utf-8"?>
<ItemAttachedData>
```



```
<Application id="Customer">
  <Data>
    <Name>John Doe</Name>
    <Company>Acme Inc</Company>
    <Email>john.doe@acme.inc</Email>
  </Data>
  <Display>Name: John Doe
Company: Acme Inc
Email: john.doe@acme.inc</Display>
</Application>
<Application id="WEBCONTEXT">
  <topic id="FirstName">
    <LABEL>FirstName</LABEL>
    <VALUE>John</VALUE>
  </topic>
  <topic id="LastName">
    <LABEL>LastName</LABEL>
    <VALUE>Doe</VALUE>
  </topic>
  <topic id="Company">
    <LABEL>Company</LABEL>
    <VALUE>Acme Inc</VALUE>
  </topic>
</Application>
</ItemAttachedData>
```

In chat there are originally two Application nodes, one for CDT (Customer) and one for IC WebClient (WEBCONTEXT). They contain basically the same information (the reason for two Application nodes is that the IC WebClient format was not known when CDT interface to chat was designed). This might change later, it would be better to use the WEBCONTEXT node only.



## 7.2.4 Item Attached Data in ICI 3.07 specification

Item attached data are relevant to communicate data between SAP applications and/or within an SAP application, and are fully visible by the contact center. The concept of the previously (SAPphone interface) known call attached data is extended for all communication channels with their items. Hence item attached data are introduced which allows adding business information to a particular item (phone call, e-mail, chat session, etc). The data are encoded in XML. Multiple applications (for example the SAP IC, the Channel Manager, an IVR, etc.) can add their specific data to this document. Every application places its data into a separate subtree. The top-level structure of the XML looks like this:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<ItemAttachedData>
  <Application id = "SAP-IC">
    ... ← SAP IC specific data will be inserted here
  </Application>
  <Application id = "IVR">
    ... ← IVR specific data will be inserted here
  </Application>
  <Application id = "...">
    ... ← Specific data of another component will be inserted here
  </Application>
</ItemAttachedData>
```

There is no XML schema definition, since it is up to each application to define the structure of its XML subtree.

Note that this is up to the SAP application to not destroy or modify the item attached data owned by other applications when performing an action on its own data.

### Example (IC WebClient):





SAP note #707104 describes how automatic account identification can be achieved through the use of item attached data. Here is the scenario: a business partner reaches the contact center through, for example, an IVR system. The business partner GUID is then retrieved from the CRM system. The Contact Center informs the SAP application (here IC WebClient) about the new phone call by firing the 'phoneCallChanged' event, with the 'attachedData' parameter already filled in the following way:

```
<Application id="CRM_IC/BUPA">
<CURRENTCUSTOMER>bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb</CURRENTCUSTOMER>
<CURRENTCONTACT>cccccccccccccccccccccccccccccccccccc</CURRENTCONTACT>
<CURRENTCOMPONENT>890FD258977D284B9EEA965DEDA369EB</CURRENTCOMPONENT>
<BPCONFIRMED></BPCONFIRMED>
</Application>
```

where bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb and ccccccccccccccccccccccccccccccccccc respectively denote the 32 character GUIDs of the customer Business Partner and of the contact person Business Partner (the latter may be omitted).

### 7.3 Action Item Routing Attributes

This is the list of supported action item attributes or routing attributes:

ATTRIBUTE NAME	DESCRIPTION
FROM_ADDRESS	Saved to Activity.SourceAddress.
TO_ADDRESS	Saved to Activity.DestinationAddress.



	Also used to find the E-Mail queue by queue address (case insensitive).
Z_TO_ADDRESS	Can be used to override TO_ADDRESS to select the E-Mail queue by queue address.
Z_TO_ADDRESS_DIS	Can be used to override TO_ADDRESS to save Activity.DestinationAddress.
SUBJECT	Saved to Activity.Subject.
SKILL	<p>Can be used to add skill requirements to Activity. The format is:</p> <p>&lt;GuidOrExtName&gt;[=1-5]</p> <p>GuidOrExtName can be either GUID of skill in Contact Center, or the external name of the skill. The required value can be given after = character, and the value can be a number from 1 to 5. If value is not given, then value is 4. There can be multiple skills (attribute can be repeated).</p>
DELETE	<p>If value is X, then existing action item is deleted (or new action item is not created). See OII.CheckDuplicateActionItem parameter for more information.</p> <p>New in CCtr 7 SP 10.</p>



REQUIRED_AGENT	Required agent GUID or CRM Login ID or User Login ID. There can be multiple required agents (attribute can be repeated).  New in CCtr 7 FP 14.
REQUIRED_AGENT_EXPIRY_TIME	Required agent expiry time in seconds. The default value is one week (604800 seconds). After the time has expired, then the action item can be allocated to any agent (who has logged into queue and has correct skills).  New in CCtr 7 FP 14.

In action item routing, there are two lists that can have these values:

`itemData.actionItemAttributes`

`routingAttributes`

Normally `itemData.actionItemAttributes` contain `FROM_ADDRESS`, `TO_ADDRESS` and `SUBJECT`, if action item comes from ERMS Push. Then `routingAttributes` can be used to override these values, by adding `Z_TO_ADDRESS` and `Z_TO_ADDRESS_DIS`, for example.

## 7.4 Knocking Calls

"Knocking" calls are not shown in IC WebClient, until the user manually picks them in CDT. Knocking calls are new incoming calls that go into user's private queue, if the user is already handling one or more active call in CDT. Normally CEM does not



allocate queue calls to user, if user is already handling some call, but it is possible that some other user transfers other queue call or makes consultation call directly to this user.

## 7.5 E-Mail Transformations

Because the current version of SAP CRM IC WebClient does not properly support all types of HTML/RTF messages, OII does the following transformations to MIME-encoded message content.

### 7.5.1 Inbound Messages

If a message contains only HTML body, OII creates a plain body text automatically. The log file contains the text **HTML converted to plain text**.

If a message contains embedded messages in the RFC-822 format, OII converts these into regular attachments. E-mail headers are in a EmailHeaderN.txt text file and the e-mail body in a EmailBodyN.txt or EmailBodyN.htm text/HTML file. The log file contains the text **EmailAttachment converted: N**.

### 7.5.2 Outbound Messages

OII adds the Case ID (CID) into the end of the subject line if it does not already contain a CID and the user is currently processing an active and accepted e-mail. The CID is copied from this active e-mail, and the subject contains the <CID0123456789AB> suffix (<CID + 12 hex characters + >). CEM uses the CID to identify the message so that it can later on offer the recipient's reply to the same user who sent the message originally. The log file contains the text **Subject changed to:**



When SAP sends a message that has attachments, the body text is in the TEXT1.TXT (in SAP CRM 4.0) or FILENAME.HTM (in SAP CRM 5.0) attachment as well. OII converts this attachment back to body text and removes the attachment. The log file contains the text **Attachment TEXT1.TXT converted to text body**.

## 7.6 SOAP Network Bandwidth

How much network bandwidth does the ICI interface require?

Different events and commands are transferred between ICI adapter (OII) and client (CRM). Changing the user status (Ready, Not Ready) sends the userChanged event, and changing the phone call status sends the phoneCallChanged event.

UserChanged event from OII: > 2000 + 315 bytes (post + response). Post size depends on the number of queues that the agent has rights to, in this example 6 queues.

In one phone call transaction, there are usually at least 2 commands (Accept and End), and 4 events (Alerting, Connected, Wrap Up, and Ended): Alerting – Accept – Connected – Wrap Up – End – Ended

One command takes  $1032 + 637 = 1700$  bytes. One event takes about  $1000 + 325 = 1500$  bytes. Total size is about 10 kB.

This amount can vary a lot. For example, if CRM saves CAD to the phone call, CAD will be transferred in each phoneCallChanged event, and the total size can easily be 15 - 20 kB.

### a. Troubleshooting with CRM ICI Trace

In addition to OII log, the CRM side has the ICI log, which can be viewed in transaction CRM\_ICI\_TRACE. By default, all users' errors are captured in the trace logs.

To capture all information, not just errors, set the user parameter CRM\_ICI\_TRACELEVEL = "XXX" via transaction SU3, or through the transaction CRM\_ICI\_TRACE itself through the drop down menu "Trace" > "Start trace user dependent".





## 8 SAPphone Functionality Overview

In 6.0 SP5 and later, the OII interface implements also certain functions of the SAPphone interface that enable the telephony control for SAP CRM/ERP applications (without ICI interface). SAPphone interface can still be used for outbound call list transfer and reporting data transfer. SAPphone interface is compatible with the SAPphone specification version 5.11.A11 (Jan 10, 2008), but it does not implement all SAPphone features.

Only phone channel is supported, so no e-mail, chat or action item channels.

### 8.1 SAPphone Server Connections

#### 8.1.1 SAP .NET Connector 2.0

Server connection string is in format:

```
-a[Program ID] -g[Gateway Host] -x[Gateway Service]
```

For example:

```
-aSAPphoneBCM -g1.2.3.4 -xsapgw76
```

There is a limit of how many SAPphone server connections can be registered. The limit is 100 in Contact Center 7 SP3, and 200 in SP4. To have more than 100 connections in SP4, the following system environment variable have to be set:

```
CPIC_MAX_CONV=200
```

After this system environment variable has been set, the IIS server (where OII is running) must be restarted.

In Contact Center 7 SP4, there is a new connection string option REG\_COUNT to make more than one server registration. For example, the following creates 20 connections:

```
-aSAPphoneBCM -g1.2.3.4 -xsapgw76 REG_COUNT=20
```

Usually there is no need to have many server connections, unless the wait-mode is used. In wait-mode, CRM/ERP calls SPS\_WAITFORCALL, and this function does not return until there is a new call for the agent. This function reserves one server



connection. So to have more than 100 agents in wait mode, there needs to be more than 100 connections.

Alternatively, the client application poll new calls by calling SPS\_GETCALLSTATE. Third option is that client application waits for SPS\_NEW\_CALL function call from Contact Center. This is a client function (SPS\_WAITFORCALL and SPS\_GETCALLSTATE are server functions).

## 8.1.2 SAP .NET Connector 3.0

Contact Center 7 SP4 also supports new SAP .NET Connector 3.0.

Also it is now possible to use .NET Connector 3.0 for server connection. This is enabled by giving the connection string in following format:

```
PROGRAM_ID=SAPphoneBCM GWHOST=uxciq0m GWSERV=sapgw76
```

.NET Connector 3.0 supports also the optional parameters REG\_COUNT and MAX\_REG\_COUNT.

.NET Connector 3.0 server connection requires that there is also client connection configured, because the server repository (function and parameter metadata) are loaded from ABAP server using the client connection. Most of the functions can be used without client connection, but the metadata is not pre-cached in OII for some rarely used functions like SPS\_GETWORKMODE, SPS\_MODIFY\_PDCALL, SPS\_DELETE\_PDCALL, SPS\_APPENDCALLINFO, SPS\_DROPPARTY.

## 8.2 SAPphone Client Connections

There are only three client functions: SPS\_NEW\_CALL, SPS\_CALL\_ENDED and SPS\_GET\_LINES\_PER\_SERVER. Contact Center calls these when there is a new call, or when call ends.

In Contact Center 7 SP4, OII uses SAP .NET Connector 3.0 for client functions. Earlier versions (and server connections still) were using version 2.0.

Connector 3.0 has the following new options for connection pool:

POOL\_SIZE





Gives the maximum number of RFC connections that this destination will keep in its pool. More connections can be opened (until MaxPoolSize is reached), but they are closed again immediately after usage.

Default is 10

#### MAX\_POOL\_SIZE

In order to prevent an unlimited number of connections to be opened (which from a certain point on would cause the entire machine's performance to deteriorate dramatically), you can set the MaxPoolSize parameter. NCo will not open further connections for this destination when this limit is reached.

Default: same value as PoolSize

These can be set in client connection string, like:

```
CLIENT=506 USER=bcm1 PASSWD=bcm1 LANG=EN ASHOST=1.2.3.4 SYSNR=20  
MAX_POOL_SIZE=100
```

If you configure more than one identical server connection string (to have more than one server connection), then there is only need to have the client connection string in the first of these. The remaining client connection strings can be left empty.

## 8.3 SAPphone Functions

Here is a list of SAPphone functions. The function is supported unless there is a comment saying otherwise. See the SAPphone specification for more information.

### 8.3.1 Call Control Functions

XCHGVERSION

XCHGPARAMS

SPS\_REGISTER

SPS\_DEREGISTER

SPS\_AGENTLOGIN

SPS\_AGENTLOGOUT

SPS\_SETWORKMODE



SPS\_GETWORKMODE  
SPS\_GETCALLSTATE  
SPS\_GETQUEUES  
SPS\_WAITFORCALL  
SPS\_CANCELWAIT  
SPS\_ANSWER  
SPS\_DEFLECT  
SPS\_MAKECALL  
SPS\_CONSULT  
SPS\_BTRANSFER  
SPS\_TRANSFER  
SPS\_CONFERENCE  
SPS\_RECONNECT  
SPS\_HOLD  
SPS\_UNHOLD  
SPS\_ALTERNATE  
SPS\_DROPPARTY  
SPS\_DROPCALL  
SPS\_DROPALL  
SPS\_GENERIC – not supported

### 8.3.2 Reporting Functions (Client Connection)

These functions are called only if the client connection is configured (and connected).

SPS\_GET\_LINES\_PER\_SERVER  
SPS\_NEW\_CALL



SPS\_CALL\_ENDED

### 8.3.3 Functions Concerning Predictive Dialing / Power Dialing

SPS\_PDLISTTRANSFER

SPS\_CAMPAIGNS\_GET – not supported

SPS\_MODIFY\_PDCALL

SPS\_DELETE\_PDCALL

SPS\_ASSIGNED\_CAMPAIGNS\_GET – not supported

SPS\_STATCAMPAIGN

### 8.3.4 Functions for Call-Attached Data from 4.5B Onwards

SPS\_GETCALLINFO

SPS\_SETCALLINFO

SPS\_APPENDCALLINFO – not supported

SPS\_DELETECALLINFO

### 8.3.5 Functions for Call-Attached Data in Interface Version

#### 4.00A

SPS\_CREATEINFO – not supported

SPS\_SETINFO – not supported

### 8.3.6 Functions for Monitoring External Components from Rel. 4.5B Onwards

SMON\_TRACE\_SET – not supported



SMON\_TRACE\_UPLOAD – not supported

SMON\_TRACE\_EXPLAIN – not supported

SMON\_COMP\_STATE\_RETRIEVE – not supported

### 8.3.7 Function for Statistical Data Upload

SPS\_STAT\_DATA\_GET – supported, this replaces the separate SAPphoneBCM application



## 9 ICI Interface Specification

Each operation consists of two messages; the request and the response.

### 9.1 IciSystem Interface

#### 9.1.1 exchangeProductInformation Operation

##### 9.1.1.1 Definition

<b>Purpose</b>	The operation is called when OII is configured to use in SAP CRM BCB.
<b>Request name</b>	exchangeProductInformation
<b>Request elements</b>	String <iciVersion> : The version of the current implementation of ICI (Integrated Communication Interface) in the SAP system (for example, 3.00).
<b>Response elements</b>	ProductInformation <response> : See the ProductInformation element for more information.

##### 9.1.1.2 exchangeProductInformation Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciSystem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciSystem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <exchangeProductInformation xmlns="urn:IciSystemInterface">
      <iciVersion xmlns="">3.00</iciVersion>
    </exchangeProductInformation>
  </soap:Body>
</soap:Envelope>
```



### 9.1.1.3 exchangeProductInformationResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <exchangeProductInformationResponse xmlns="urn:IciSystemInterface">
      <response>
        <message xsi:nil="true" xmlns="" />
        <productName xmlns="">WicomOII</productName>
        <iciMessageGroups xmlns="">
          <item>IciGroupBasic</item>
          <item>IciGroupUser</item>
          <item>IciGroupTelephony</item>
          <item>IciGroupMessaging</item>
        </iciMessageGroups>
        <productVersion xmlns="">1.0.2214.21309</productVersion>
        <iciVersion xmlns="">3.00</iciVersion>
      </response>
    </exchangeProductInformationResponse>
  </soap:Body>
</soap:Envelope>
```

### 9.1.2 getWorkcenterCapability Operation

#### 9.1.2.1 Definition

<b>Purpose</b>	The operation is called first when user logs in using SAP CRM 4.1 or later (4.0 didn't use this function). OII always responds with the same values, because Free Seating feature is not supported.
<b>Request name</b>	getWorkcenterCapability
<b>Request elements</b>	String <userId> : The user in question.
<b>Response elements</b>	String <types> : Always 0. Workcenter identification types used by the contact center to identify workcenters. A string of comma-separated workcenter identification types (for instance: "1,2,3,100"). Position of a type in the string gives higher priority to the type relative to other types that follow it in the string.



	<p>0: Free Seating not supported. User-based workcenter identification.</p> <p>1: Workcenter identified by a fully qualified domain name (e.g. raptor.wdf.sap.corp)</p> <p>2: Workcenter identified by a hostname (e.g. raptor)</p> <p>3: Workcenter identified by an IP address (e.g. 10.12.124.252)</p> <p>&gt;=100: Workcenter identified by unspecified ID.</p> <p>Boolean &lt;workcenterList&gt; : Always false. Informs if workcenter lists are supported. It is not mandatory for the contact center to supply any value in this parameter. If the parameter is not supplied, then the default value is assumed that is 'false'.</p> <p>Boolean &lt;filter&gt; : Always false. Informs if workcenter list filtering is supported. The parameter is used only if the workcenter list parameter value returned is true. It is not mandatory for the contact center to supply any value in this parameter. If the parameter is not supplied, then the default value is assumed that is 'false'.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 9.1.2.2 getWorkcenterCapability Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciSystem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciSystem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <getWorkcenterCapability xmlns="urn:IciSystemInterface">
      <userId>string</userId>
    </getWorkcenterCapability>
  </soap:Body>
</soap:Envelope>

```



### 9.1.2.3 getWorkcenterCapabilityResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getWorkcenterCapabilityResponse xmlns="urn:IciSystemInterface">
      <response>
        <types>string</types>
        <workcenterList>boolean</workcenterList>
        <filter>boolean</filter>
      </response>
    </getWorkcenterCapabilityResponse>
  </soap:Body>
</soap:Envelope>
```

### 9.1.3 getPresenceQueueInfo Operation

#### 9.1.3.1 Definition

<b>Purpose</b>	Retrieves presence queue data for a specific channel. Used in CRM 2006 (5.2) and later.
<b>Request name</b>	getPresenceQueueInfo
<b>Request elements</b>	<p><b>String &lt;userId&gt;</b> : User who requests the presence info.</p> <p><b>String &lt;channelType&gt;</b> :</p> <ul style="list-style-type: none"> <li>1 = telephony</li> <li>2 = messaging</li> <li>3 = chat</li> <li>4 = action routing</li> </ul> <p>There seems to be a bug in CRM 5.0 – 7.0 because it always asks presence for telephony channel queues even if an e-mail or chat is being processed in IC WebClient.</p>
<b>Response elements</b>	<p><b>GroupInfo[] &lt;groupInfo&gt;</b> : List of groups into which the queues fall into. See the GroupInfo element for more information. Currently always returns only one group:</p> <p>id=1 description=*</p> <p><b>QueueInfo[] &lt;queueInfo&gt;</b> : List of queues the user is permitted to transfer to. See the QueueInfo element for more information. A list of queues that the user has right to see:</p>





	<p>id=Queue address (phone number or e-mail address).  description=Queue name.  loggedIn=Number of logged in agents.  ready=Number of free agents.  pending=Number of queuing items.  open=Is queue open = 1, closed = 0, empty = no info.  avgWaitingTime=Average waiting time in seconds.  maxWaitingTime=Maximum waiting time in seconds.  busy=Number of busy agents.  wrapUp=Number of wrap up agents.  notReady=Number of not ready agents.  paused=Number of paused agents.</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 9.1.3.2 getPresenceQueueInfo Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciSystemInterface">SystemLanguage</language>
    <user xmlns="urn:IciSystemInterface">SystemUser</user>
  </soap:Header>
  <soap:Body>
    <getPresenceQueueInfo xmlns="urn:IciSystemInterface">
      <userId xmlns="urn:IciUserInterface">string</userId>
      <channelType>string</channelType>
    </getPresenceQueueInfo>
  </soap:Body>
</soap:Envelope>
```

### 9.1.3.3 getPresenceQueueInfoResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getPresenceQueueInfoResponse xmlns="urn:IciSystemInterface">
      <response>
        <groupInfo>
          <item>
            <id>string</id>
            <description>string</description>
            <parentId>string</parentId>
            <queuesId xsi:nil="true" />
          </item>
          <item>
            <id>string</id>
            <description>string</description>
            <parentId>string</parentId>
            <queuesId xsi:nil="true" />
          </item>
        </groupInfo>
        <queueInfo>
          <item>
```



```

<id>string</id>
<description>string</description>
<guid>string</guid>
<loggedIn>string</loggedIn>
<ready>string</ready>
<pending>string</pending>
<open>string</open>
<avgWaitingTime>string</avgWaitingTime>
<maxWaitingTime>string</maxWaitingTime>
<busy>string</busy>
<wrapUp>string</wrapUp>
<notReady>string</notReady>
<paused>string</paused>
</item>
<item>
  <id>string</id>
  <description>string</description>
  <guid>string</guid>
  <loggedIn>string</loggedIn>
  <ready>string</ready>
  <pending>string</pending>
  <open>string</open>
  <avgWaitingTime>string</avgWaitingTime>
  <maxWaitingTime>string</maxWaitingTime>
  <busy>string</busy>
  <wrapUp>string</wrapUp>
  <notReady>string</notReady>
  <paused>string</paused>
</item>
</queueInfo>
</response>
</getPresenceQueueInfoResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.1.4 getUserPresence Operation

### 9.1.4.1 Definition

<b>Purpose</b>	Retrieves presence data for specified users. Used in CRM 7.0.
<b>Request name</b>	getUserPresence
<b>Request elements</b>	<p><b>String[] &lt;users&gt;</b> : List of user ids for which the presence should be returned.</p> <p><b>String &lt;searchterm&gt;</b> : A free-text search term for users. Can contain user first name or surname or both separated by space, or user title or address, depending on what information is stored in Contact Center directory.</p> <p><b>String &lt;maxhits&gt;</b> : Integer specifying the maximum number of users that should be returned. A default value of 100 is used. Used only when searchterm is used.</p>



	The request can contain either <b>users</b> or <b>searchterm</b> , and the searchterm is used only if users list is empty.
<b>Response elements</b>	<b>ici.UserPresenceInfo [] &lt;userPresenceInfo&gt;</b> : List of users and their presence information, including phone numbers. See the UserPresenceInfo element for more information.

### 9.1.4.2 getUserPresence Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciSystemInterface">SystemLanguage</language>
    <user xmlns="urn:IciSystemInterface">SystemUser</user>
  </soap:Header>
  <soap:Body>
    <getUserPresence xmlns="urn:IciSystemInterface">
      <users>
        <Item>string</Item>
        <Item>string</Item>
      </users>
      <searchterm>string</searchterm>
      <maxhits>string</maxhits>
    </getUserPresence>
  </soap:Body>
</soap:Envelope>
```

### 9.1.4.3 getUserPresenceResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getUserPresenceResponse xmlns="urn:IciSystemInterface">
      <response>
        <item>
          <user>string</user>
          <presenceStatus>string</presenceStatus>
          <presenceDescription>string</presenceDescription>
          <collaborationData>
            <item xsi:nil="true" />
            <item xsi:nil="true" />
          </collaborationData>
        </item>
        <item>
          <user>string</user>
          <presenceStatus>string</presenceStatus>
          <presenceDescription>string</presenceDescription>
          <collaborationData>
            <item xsi:nil="true" />
            <item xsi:nil="true" />
          </collaborationData>
        </item>
      </response>
    </getUserPresenceResponse>
  </soap:Body>
</soap:Envelope>
```



```

        </response>
    </getUserPresenceResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.1.5 getCallRecordings Operation

### 9.1.5.1 Definition

<p><b>Purpose</b></p>	<p>Retrieves call recording file name. New experimental function in Contact Center 7 SP4, not supported in CRM except by customization.</p> <p>Requires the following registry settings under virtual unit:</p> <p>REPORTING_DSN – SQL data source name for reporting database.</p> <p>MONITORING_DSN – SQL data source name for monitoring database.</p> <p>Reporting and monitoring databases are used to get the following response elements: startTime, endTime, source, destination.</p>
<p><b>Request name</b></p>	<p>getCallRecordings</p>
<p><b>Request elements</b></p>	<p><b>String &lt;callHandle&gt;</b> : Search recordings by call handle.</p> <p><b>String &lt;userId&gt;</b> : Search recordings by user ID.</p> <p><b>String &lt;dateTimeFrom&gt;</b> : Search recordings by timestamp, lower limit.</p> <p><b>String &lt;dateTimeTo&gt;</b> : Search recordings by timestamp, upper limit.</p> <p><b>String &lt;filePath&gt;</b> : Search recordings by file path.</p> <p><b>String &lt;fileName&gt;</b> : Search recordings by file name.</p> <p><b>String &lt;maxhits&gt;</b> : Integer specifying the maximum number of call recordings that should be returned. Default is to return all.</p>
<p><b>Response elements</b></p>	<p><b>ici.CallRecording [] &lt;getCallRecordingsResponse&gt;</b> : List of call recordings. See the CallRecording element for more information.</p>



### 9.1.5.2 getCallRecordings Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciSystemInterface">SystemLanguage</language>
    <user xmlns="urn:IciSystemInterface">SystemUser</user>
  </soap:Header>
  <soap:Body>
    <getCallRecordings xmlns="urn:IciSystemInterface">
      <callHandle>string</callHandle>
      <userId>string</userId>
      <dateTimeFrom>string</dateTimeFrom>
      <dateTimeTo>string</dateTimeTo>
      <filePath>string</filePath>
      <fileName>string</fileName>
      <maxhits>string</maxhits>
    </getCallRecordings>
  </soap:Body>
</soap:Envelope>
```

### 9.1.5.3 getCallRecordingsResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getCallRecordingsResponse xmlns="urn:IciSystemInterface">
      <response>
        <item>
          <callHandle>string</callHandle>
          <userId>string</userId>
          <creationTime>dateTime</creationTime>
          <startTime>dateTime</startTime>
          <endTime>dateTime</endTime>
          <source>string</source>
          <destination>string</destination>
          <fileName>string</fileName>
        </item>
        <item>
          <callHandle>string</callHandle>
          <userId>string</userId>
          <creationTime>dateTime</creationTime>
          <startTime>dateTime</startTime>
          <endTime>dateTime</endTime>
          <source>string</source>
          <destination>string</destination>
          <fileName>string</fileName>
        </item>
      </response>
    </getCallRecordingsResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.1.6 createCallBack Operation

Used for the web callback function, added in 7.0 SP07.

### 9.1.6.1 Definition

<b>Purpose</b>	Creates a callback request. New function in Contact Center 7 SP7, not supported in CRM.
<b>Request name</b>	createCallBack
<b>Request elements</b>	<p><b>String &lt;number&gt;</b> : The phone number of the customer.</p> <p><b>String &lt;queue&gt;</b> : Queue number, or queue GUID, or queue name. The queue should have call back script configured or no script at all.</p> <p><b>String &lt;callTime&gt;</b> : callTime: Optional date and time when call should be made. Various formats are allowed, including YYYY-MM-DD HH:MM:SS. Time is in UTC.</p> <p><b>String &lt;extraData&gt;</b> : Optional extra data in XML format. The root node must be named "XML".</p> <p><b>String &lt;gwPrefix:&gt;</b> : Optional gateway prefix, used in call routing.</p> <p><b>String &lt;notes&gt;</b> : Optional notes for the call back. This is shown in call back script.</p>
<b>Response elements</b>	<b>String &lt;createCallBackResponse&gt;</b> : GUID of the call back if operation succeeded. NULL if there was already call back created for the same number in same queue.

### 9.1.6.2 extraData XML Format

Language as skill (or any other skill requirements for that matter), contact priority, as well as basically any arbitrary "CAD" key value pair can all be passed to CEM by utilizing the extraData XML string parameter.



## EXAMPLES

### Skills

```
<XML><Skills>GuidOrExtName=value;GuidOrExtName=value</Skills></XML>
```

...where GuidOrExtName is skill GUID or external name, and value is an integer between (and including) 1-5. If there is invalid GuidOrExtName, then it is dropped by OII and error printed to log (operation is not aborted).

### Priority

```
<XML><Priority>12345</Priority></XML>
```

Any positive integer value will do. Invalid values (less than or equal to zero, or something that won't convert to an integer) result in value 1 being used as contact priority.

### "Any" CAD

```
<XML><Foo>Bar</Foo></XML>
```

Preferred agent information cannot be attached to callback requests.

## 9.1.6.3 createCallback Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciSystemInterface">SystemLanguage</language>
    <user xmlns="urn:IciSystemInterface">SystemUser</user>
  </soap:Header>
  <soap:Body>
    <createCallback xmlns="urn:IciSystemInterface">
      <number>string</number>
      <queue>string</queue>
      <callTime>string</callTime>
      <extraData>string</extraData>
      <gwPrefix>string</gwPrefix>
      <notes>string</notes>
    </createCallback>
  </soap:Body>
</soap:Envelope>
```

## 9.1.6.4 createCallbackResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createCallbackResponse xmlns="urn:IciSystemInterface">
```



```
<response>string</response>
</createCallBackResponse>
</soap:Body>
</soap:Envelope>
```

## 9.1.7 updateCallBack Operation

Used for the web callback function, added in 7.0 SP07.

### 9.1.7.1 Definition

<b>Purpose</b>	Updates callback request. New function in Contact Center 7 SP7, not supported in CRM.
<b>Request name</b>	updateCallBack
<b>Request elements</b>	<p><b>String &lt;id&gt;</b> : GUID of the call back, returned by createCallBack or getCallBacks.</p> <p><b>String &lt;result&gt;</b> : One of the failures: BADADDRESS, BUSY, NO_ANSWER, VOICEMAIL, OTHER                  These will increment Failures count, until it reaches MaxCalls, in which case result will be HANDLED.</p> <p>Or success: SUCCESSFUL</p> <p>Or remove call back: HANDLED</p> <p><b>String &lt;notes&gt;</b> : Optional notes for the callback, shown in the callback script. NULL value is ignored.</p>
<b>Response elements</b>	<b>void &lt;updateCallBackResponse&gt;</b> : No data.

### 9.1.7.2 updateCallBack Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciSystemInterface">SystemLanguage</language>
    <user xmlns="urn:IciSystemInterface">SystemUser</user>
```





```

</soap:Header>
<soap:Body>
  <updateCallBack xmlns="urn:IciSystemInterface">
    <id>string</id>
    <result>string</result>
    <notes>string</notes>
  </updateCallBack>
</soap:Body>
</soap:Envelope>
    
```

### 9.1.7.3 updateCallBackResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateCallBackResponse xmlns="urn:IciSystemInterface" />
  </soap:Body>
</soap:Envelope>
    
```

## 9.1.8 getCallBacks Operation

Used for the web callback function, added in 7.0 SP07.

### 9.1.8.1 Definition

<b>Purpose</b>	Returns all callbacks. New function in Contact Center 7 SP7, not supported in CRM.
<b>Request name</b>	getCallBacks
<b>Request elements</b>	None.
<b>Response elements</b>	<b>ici.CallBack [] &lt;getCallBacksResponse&gt;</b> : List of call backs. See the CallBack element for more information.

### 9.1.8.2 getCallBacks Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    
```



```

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Header>
    <language xmlns="urn:IciSystemInterface">SystemLanguage</language>
    <user xmlns="urn:IciSystemInterface">SystemUser</user>
  </soap:Header>
  <soap:Body>
    <getCallbacks xmlns="urn:IciSystemInterface" />
  </soap:Body>
</soap:Envelope>

```

### 9.1.8.3 getCallbacksResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getCallbacksResponse xmlns="urn:IciSystemInterface">
      <response>
        <item>
          <id>string</id>
          <number>string</number>
          <queueNumber>string</queueNumber>
          <handledTime>dateTime</handledTime>
          <lastCallTime>dateTime</lastCallTime>
          <nextCallTime>dateTime</nextCallTime>
          <failures>int</failures>
          <maxCalls>int</maxCalls>
          <lastResult>string</lastResult>
          <extraData>string</extraData>
          <notes>string</notes>
          <gwPrefix>string</gwPrefix>
        </item>
        <item>
          <id>string</id>
          <number>string</number>
          <queueNumber>string</queueNumber>
          <handledTime>dateTime</handledTime>
          <lastCallTime>dateTime</lastCallTime>
          <nextCallTime>dateTime</nextCallTime>
          <failures>int</failures>
          <maxCalls>int</maxCalls>
          <lastResult>string</lastResult>
          <extraData>string</extraData>
          <notes>string</notes>
          <gwPrefix>string</gwPrefix>
        </item>
      </response>
    </getCallbacksResponse>
  </soap:Body>
</soap:Envelope>

```



## 9.2 IciUser Interface

### 9.2.1 getAttributes Operation

#### 9.2.1.1 Definition

<b>Purpose</b>	The operation is called when a user logs into SAP CRM IC WebClient. OII updates the user information from the database, and if the ICIUSER_AUTOMATIC_UNSUBSCRIBE setting is true, all user subscriptions are unsubscribed (because sometimes certain SAP versions did not do it automatically).
<b>Request name</b>	getAttributes
<b>Request elements</b>	<p><b>String &lt;userId&gt;</b> : Identifies the user. Must match the user ID in the Contact Center system.</p> <p><b>String &lt;cmsPing&gt;</b> : New optional parameter in Contact Center 7 SP4. If value equals TRUE, then CRM is just testing if CMS (Contact Center) is alive. Response will be empty in this case, except for the new value cmsHostId, which identifies CMS system. It contains the value of OII module GUID. cmsHostId is filled always, even if cmsPing is different than TRUE.</p>
<b>Response elements</b>	<b>User &lt;response&gt;</b> : See the User element for more information.

#### 9.2.1.2 getAttributes Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
  </soap:Header>
  <soap:Body>
    <getAttributes xmlns="urn:IciUserInterface">
      <userId xmlns="">string</userId>
    </getAttributes>
  </soap:Body>
```



```
</soap:Envelope>
```

### 9.2.1.3 getAttributesResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttributesResponse xmlns="urn:IciUserInterface">
      <response>
        <userId>HARRI</userId>
        <currentWorkmode>
          <id>1</id>
          <description>Logged off</description>
        </currentWorkmode>
        <workmodes>
          <item>
            <id>1</id>
            <description>Logged off</description>
          </item>
          <item>
            <id>2</id>
            <description>Logged on - ready</description>
          </item>
          <item>
            <id>3</id>
            <description>Logged on - not ready</description>
          </item>
          <item>
            <id>1000</id>
            <description>No Phone</description>
          </item>
        </workmodes>
        <wrapUpMode>1</wrapUpMode>
        <currentChannels>
        </currentChannels>
        <channels>
          <item>
            <id>1</id>
            <description>Telephony</description>
          </item>
        </channels>
        <currentQueues>
          <item>
            <id>2001</id>
            <description>Q 2001</description>
          </item>
          <item>
            <id>2003</id>
            <description>Q 2003</description>
          </item>
        </currentQueues>
        <queues>
          <item>
            <id>2001</id>
            <description>Q 2001</description>
          </item>
          <item>
            <id>2002</id>
            <description>Q 2002</description>
          </item>
        </queues>
      </response>
    </getAttributesResponse>
  </soap:Body>
</soap:Envelope>
```



```

        </item>
        <item>
            <id>2003</id>
            <description>Q 2003</description>
        </item>
    </queues>
    <addresses>
        <item>
            <address>+129</address>
            <channel>1</channel>
        </item>
    </addresses>
</response>
<cmsHostId>BA48C51B1B9249F6A56C4E6ADFFC2E3F</cmsHostId>
</getAttributesResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.2.2 setCurrentWorkmode Operation

### 9.2.2.1 Definition

<b>Purpose</b>	The operation sets the workmode for a user.
<b>Request name</b>	setCurrentWorkmode
<b>Request elements</b>	<p><b>String &lt;userId&gt;</b> : Identifies the user.</p> <p><b>String &lt;workmode&gt;</b> : The workmode to be set.</p>
<b>Response elements</b>	(no parameters)

### 9.2.2.2 setCurrentWorkmode Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
  </soap:Header>
  <soap:Body>
    <setCurrentWorkmode xmlns="urn:IciUserInterface">
      <userId xmlns="">HARRI</userId>
      <workmode xmlns="">1</workmode>
    </setCurrentWorkmode>
  </soap:Body>
</soap:Envelope>
    
```



### 9.2.2.3 setCurrentWorkmodeResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setCurrentWorkmodeResponse xmlns="urn:IciUserInterface">
      <response />
    </setCurrentWorkmodeResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.2.3 setCurrentChannels Operation

### 9.2.3.1 Definition

<b>Purpose</b>	The operation defines the information about the channels that are enabled or disabled for the user. Currently this operation does nothing in CCtr.
<b>Request name</b>	setCurrentChannels
<b>Request elements</b>	<b>String &lt;userId&gt;</b> : Identifies the user. <b>String[] &lt;channels&gt;</b> : The channels that are enabled for this user.
<b>Response elements</b>	(no parameters)

### 9.2.3.2 setCurrentChannels Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
  </soap:Header>
  <soap:Body>
    <setCurrentChannels xmlns="urn:IciUserInterface">
      <userId xmlns="">HARRI</userId>
      <channels xmlns="">
        <item>1</item>
      </channels>
    </setCurrentChannels>
  </soap:Body>
</soap:Envelope>
```



### 9.2.3.3 setCurrentChannelsResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setCurrentChannelsResponse xmlns="urn:IciUserInterface">
      <response />
    </setCurrentChannelsResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.2.4 setCurrentQueues Operation

### 9.2.4.1 Definition

<b>Purpose</b>	The operation sets the queues the user must be assigned to. If the ICIUSER_SETCURRENTQUEUES setting is disabled (as it is by default), this function does nothing. In that case SAP has no control over the queues in CDT and the user activates and deactivates the queues.
<b>Request name</b>	setCurrentQueues
<b>Request elements</b>	<b>String &lt;userId&gt;</b> : Identifies the user. <b>String[] &lt;queues&gt;</b> : The list of queues the user is assigned to.
<b>Response elements</b>	(no parameters)

### 9.2.4.2 setCurrentQueues Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
  </soap:Header>
```



```
<soap:Body>
  <setCurrentQueues xmlns="urn:IciUserInterface">
    <userId xmlns="">HARRI</userId>
    <queues xmlns="">
      <item>2001</item>
      <item>2003</item>
    </queues>
  </setCurrentQueues>
</soap:Body>
</soap:Envelope>
```

### 9.2.4.3 setCurrentQueuesResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setCurrentQueuesResponse xmlns="urn:IciUserInterface">
      <response />
    </setCurrentQueuesResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.2.5 setAddresses Operation

### 9.2.5.1 Definition

<b>Purpose</b>	The operation sets the channel-specific addresses for the user.
<b>Request name</b>	setAddresses
<b>Request elements</b>	<b>String &lt;userId&gt;</b> : Identifies the user. <b>Address[] &lt;addresses&gt;</b> : The list of addresses for the user.
<b>Response elements</b>	(no parameters)

### 9.2.5.2 setAddresses Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
```





```

</soap:Header>
<soap:Body>
  <setAddresses xmlns="urn:IciUserInterface">
    <userId xmlns="">HARRI</userId>
    <addresses xmlns="">
      <item>
        <address>+123</address>
        <channel>1</channel>
      </item>
      <item>
        <address>name@domain.com</address>
        <channel>2</channel>
      </item>
    </addresses>
  </setAddresses>
</soap:Body>
</soap:Envelope>
    
```

### 9.2.5.3 setAddressesResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setAddressesResponse xmlns="urn:IciUserInterface">
      <response />
    </setAddressesResponse>
  </soap:Body>
</soap:Envelope>
    
```

## 9.2.6 subscribe Operation

### 9.2.6.1 Definition

<b>Purpose</b>	The operation subscribes an application to user events. The subscription key consists of the appUrl and appld, i.e. the application may send multiple subscribes with the same appUrl but different applds. In that case the contact center may collect events of the same type and content.
<b>Request name</b>	Subscribe
<b>Request elements</b>	<p><b>String &lt;appUrl&gt;</b> : Identifies the application and represents the address of the SOAP entry point to which the events have to be sent to. The format follows the URL rules, for example http://sapserver:80/soapdispatcher.</p> <p><b>String &lt;appld&gt;</b> : Identifies the subscribing application instance.</p>



	<b>String &lt;userId&gt;</b> : The user ID of the user to be monitored.
<b>Response elements</b>	(no parameters)

### 9.2.6.2 subscribe Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
  </soap:Header>
  <soap:Body>
    <subscribe xmlns="urn:IciUserInterface">
      <appUrl xmlns="">
http://ip:50000/IciEventService/IciEventConf?style=document</appUrl>
      <appId xmlns="">lcb65470-a2c1-11da-aa97-001143dc2e6e</appId>
      <userId xmlns="">HARRI</userId>
    </subscribe>
  </soap:Body>
</soap:Envelope>
```

### 9.2.6.3 subscribeResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <subscribeResponse xmlns="urn:IciUserInterface">
      <response />
    </subscribeResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.2.7 unsubscribe Operation

### 9.2.7.1 Definition

<b>Purpose</b>	The application must call this operation if it does not want to receive events anymore.
----------------	-----------------------------------------------------------------------------------------



<b>Request name</b>	Unsubscribe
<b>Request elements</b>	<p><b>String &lt;appUrl&gt;</b> : Identifies the application and represents the address of the SOAP entry point to which the events have to be sent to. The format follows the URL rules, for example http://sapserver:80/soapdispatcher.</p> <p><b>String &lt;appld&gt;</b> : Identifies the subscribing application instance.</p>
<b>Response elements</b>	(no parameters)

### 9.2.7.2 unsubscribe Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciUserInterface">UserLanguage</language>
    <user xmlns="urn:IciUserInterface">UserUser</user>
  </soap:Header>
  <soap:Body>
    <unsubscribe xmlns="urn:IciUserInterface">
      <appUrl xmlns="">
http://ip:50000/IciEventService/IciEventConf?style=document</appUrl>
      <appId xmlns="">1cb65470-a2c1-11da-aa97-001143dc2e6e</appId>
    </unsubscribe>
  </soap:Body>
</soap:Envelope>
```

### 9.2.7.3 unsubscribeResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <unsubscribeResponse xmlns="urn:IciUserInterface">
      <response />
    </unsubscribeResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.3 IciEvent Interface

### 9.3.1 userChanged Operation

#### 9.3.1.1 Definition

<b>Purpose</b>	The operation signals a change in the user’s properties (for example, in the current workmode, or in the assigned channels or queues).
<b>Request name</b>	userChanged
<b>Request elements</b>	<b>String[] &lt;appIds&gt;</b> : Contains at least one appId which has subscribed for this event.  <b>User &lt;user&gt;</b> : The user’s attributes.
<b>Response elements</b>	(no parameters)

#### 9.3.1.2 userChanged Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <userChanged xmlns="urn:IciEventInterface">
      <appIds>
        <item>1cb65470-a2c1-11da-aa97-001143dc2e6e</item>
      </appIds>
      <user>
        <userId>HARRI</userId>
        <currentWorkmode>
          <id>1</id>
          <description>Logged off</description>
        </currentWorkmode>
        <workmodes>
          <item>
            <id>1</id>
            <description>Logged off</description>
          </item>
          <item>
            <id>2</id>
            <description>Logged on - ready</description>
          </item>
        </workmodes>
      </user>
    </userChanged>
  </soap:Body>
</soap:Envelope>
```



```

    <item>
      <id>3</id>
      <description>Logged on - not ready</description>
    </item>
    <item>
      <id>1000</id>
      <description>No Phone</description>
    </item>
  </workmodes>
  <wrapUpMode>1</wrapUpMode>
  <currentChannels />
  <channels>
    <item>
      <id>1</id>
      <description>Telephony</description>
    </item>
    <item>
      <id>2</id>
      <description>E-mail</description>
    </item>
  </channels>
  <currentQueues>
    <item>
      <id>2001</id>
      <description>Q 2001</description>
    </item>
  </currentQueues>
  <queues>
    <item>
      <id>2001</id>
      <description>Q 2001</description>
    </item>
  </queues>
  <addresses>
    <item>
      <address>+123</address>
      <channel>1</channel>
    </item>
  </addresses>
</user>
</userChanged>
</soap:Body>
</soap:Envelope>

```

### 9.3.1.3 userChangedResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <userChangedResponse xmlns="urn:IciEventInterface">
      <response />
    </userChangedResponse>
  </soap:Body>
</soap:Envelope>

```



## 9.3.2 subscriptionEnded Operation

### 9.3.2.1 Definition

<b>Purpose</b>	The operation-related event is sent by the contact center when a previously accepted subscription request was canceled.
<b>Request name</b>	subscriptionEnded
<b>Request elements</b>	<p><b>String[] &lt;applds&gt;</b> : Contains at least one appld which has subscribed for this event.</p> <p><b>TextElement &lt;reason&gt;</b> : One of the following values:</p> <p>1 = Contact center shut-down phase. Subscriptions will be lost and resubscriptions are necessary.</p> <p>2 = Contact center paused. The contact center is out of service for an unspecified time. Subscriptions will not be lost and a resubscription is not necessary.</p> <p>3 = Subscription timeout. The application needs a resubscription if it is still interested in getting the events.</p>
<b>Response elements</b>	(no parameters)

### 9.3.2.2 subscriptionEnded Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <subscriptionEnded xmlns="urn:IciEventInterface">
      <applds>
        <item>1cb65470-a2c1-11da-aa97-001143dc2e6e</item>
      </applds>
      <reason>
        <id>1</id>
        <description>WicomOII Stop</description>
      </reason>
    </subscriptionEnded>
  </soap:Body>
</soap:Envelope>
```



### 9.3.2.3 subscriptionEndedResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <subscriptionEndedResponse xmlns="urn:IciEventInterface">
      <response />
    </subscriptionEndedResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.3.3 containerChanged Operation

### 9.3.3.1 Definition

<b>Purpose</b>	The operation-related event is sent by the contact center if the status of a container has changed.
<b>Request name</b>	containerChanged
<b>Request elements</b>	<p><b>String[] &lt;applds&gt;</b> : Contains at least one appld which has subscribed for this event.</p> <p><b>String &lt;channelType&gt;</b> : The container type of this event:</p> <ul style="list-style-type: none"> <li>1 = Phone line</li> <li>2 = Folder</li> <li>3 = Chat line</li> <li>4 = Action item inbox</li> </ul> <p><b>String &lt;containerId&gt;</b> : The ID of the container that has changed.</p> <p><b>TextElement &lt;containerStatus&gt;</b> : One of the following values:</p> <ul style="list-style-type: none"> <li>1 = OK</li> <li>2 = Failure</li> <li>3 = Removed</li> </ul>



<b>Response elements</b>	(no parameters)
--------------------------	-----------------

### 9.3.3.2 containerChanged Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <containerChanged xmlns="urn:IciEventInterface">
      <appIds>
        <item>string</item>
      </appIds>
      <channelType>string</channelType>
      <containerId>string</containerId>
      <containerStatus>
        <id>string</id>
        <description>string</description>
      </containerStatus>
    </containerChanged>
  </soap:Body>
</soap:Envelope>
```

### 9.3.3.3 containerChangedResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <containerChangedResponse xmlns="urn:IciEventInterface">
      <response />
    </containerChangedResponse>
  </soap:Body>
</soap:Envelope>
```





## 9.3.4 phoneCallChanged Operation

### 9.3.4.1 Definition

<b>Purpose</b>	The operation signals a change of a phone call and delivers new phone call data. It is sent for new phone calls as well.
<b>Request name</b>	phoneCallChanged
<b>Request elements</b>	<p><b>String[] &lt;appIds&gt;</b> : Contains at least one appId which has subscribed for this event.</p> <p><b>PhoneCall &lt;phoneCall&gt;</b> : The phone call data.</p>
<b>Response elements</b>	(no parameters)

### 9.3.4.2 phoneCallChanged Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <phoneCallChanged xmlns="urn:IciEventInterface">
      <appIds>
        <item>string</item>
      </appIds>
      <phoneCall xmlns="">
        <lineNumber>string</lineNumber>
        <callHandle>string</callHandle>
        <processingStatus>
          <id>string</id>
          <description>string</description>
        </processingStatus>
        <remoteNumbers>
          <item>string</item>
        </remoteNumbers>
        <dialedNumber>string</dialedNumber>
        <callStatus>
          <id>string</id>
          <description>string</description>
        </callStatus>
        <capabilityList>
          <item>int</item>
          <item>int</item>
        </capabilityList>
    </phoneCallChanged>
  </soap:Body>
</soap:Envelope>
```



```

        <inbound>boolean</inbound>
        <internal>boolean</internal>
        <attachedData>string</attachedData>
        <trunkId>string</trunkId>
    </phoneCall>
</phoneCallChanged>
</soap:Body>
</soap:Envelope>
    
```

### 9.3.4.3 phoneCallChangedResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <phoneCallChangedResponse xmlns="urn:IciEventInterface">
      <response />
    </phoneCallChangedResponse>
  </soap:Body>
</soap:Envelope>
    
```

## 9.3.5 messageChanged Operation

### 9.3.5.1 Definition

<b>Purpose</b>	The operation signals a change of a message and delivers some new small-sized data (i.e., usually no message contents). It is sent for new and deleted messages as well.
<b>Request name</b>	messageChanged
<b>Request elements</b>	<p><b>String[] &lt;appls&gt;</b> : Contains at least one appld which has subscribed for this event.</p> <p><b>Message &lt;message&gt;</b> : The new, changed or deleted message.</p>
<b>Response elements</b>	(no parameters)

### 9.3.5.2 messageChanged Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    
```



```

    <language
xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string1</language>
  <user xmlns="http://wicom.com/ws/OII/IciUser/literalTypes">string</user>
</soap:Header>
<soap:Body>
  <messageChanged xmlns="urn:IciEventInterface">
    <appIds>
      <item>string</item>
      <item>string</item>
    </appIds>
    <message>
      <folder>string</folder>
      <subject>string</subject>
      <messageId>string</messageId>
      <processingStatus>
        <id>string</id>
        <description>string</description>
      </processingStatus>
      <originator>string</originator>
      <messageText>string</messageText>
      <attachments>
        <item>string</item>
        <item>string</item>
      </attachments>
      <messageStatus>
        <id>string</id>
        <description>string</description>
      </messageStatus>
      <capabilityList>
        <item>int</item>
        <item>int</item>
      </capabilityList>
      <attachedData>string</attachedData>
      <originalMessageId>string</originalMessageId>
      <notification>boolean</notification>
      <read>boolean</read>
      <answered>boolean</answered>
      <forwarded>boolean</forwarded>
    </message>
  </messageChanged>
</soap:Body>
</soap:Envelope>

```

### 9.3.5.3 messageChangedResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <messageChangedResponse xmlns="urn:IciEventInterface">
      <response />
    </messageChangedResponse>
  </soap:Body>
</soap:Envelope>

```



## 9.3.6 chatChanged Operation

### 9.3.6.1 Definition

<b>Purpose</b>	Signals a change of a chat session. It is sent for new chat sessions as well.
<b>Request name</b>	chatChanged
<b>Request elements</b>	<p><b>String[] &lt;appIds&gt;</b> : Contains at least one appId which has subscribed for this event.</p> <p><b>ChatSession &lt;chatSession&gt;</b> : Changed chat session.</p>
<b>Response elements</b>	(no parameters)

### 9.3.6.2 chatChanged Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciEventInterface">EventLanguage</language>
    <user xmlns="urn:IciEventInterface">EventUser</user>
  </soap:Header>
  <soap:Body>
    <chatChanged xmlns="urn:IciEventInterface">
      <appIds>
        <item>string</item>
        <item>string</item>
      </appIds>
      <chatSession>
        <title>string</title>
        <chatLineId>string</chatLineId>
        <chatSessionId>string</chatSessionId>
        <processingStatus>
          <id>string</id>
          <description>string</description>
        </processingStatus>
        <capabilityList>
          <item>int</item>
          <item>int</item>
        </capabilityList>
        <attachedData>string</attachedData>
        <chatStatus>
          <id>string</id>
          <description>string</description>
        </chatStatus>
        <chatParticipants>
          <item>string</item>
          <item>string</item>
        </chatParticipants>
      </chatChanged>
    </soap:Body>
  </soap:Envelope>
```



```

        </chatSession>
    </chatChanged>
</soap:Body>
</soap:Envelope>
    
```

### 9.3.6.3 chatChangedResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <chatChangedResponse xmlns="urn:IciEventInterface">
      <response />
    </chatChangedResponse>
  </soap:Body>
</soap:Envelope>
    
```

### 9.3.7 chatNewPosting Operation

#### 9.3.7.1 Definition

<b>Purpose</b>	Signals a new posting in a chat session. It must be sent for all subscribed lines, also for the one which has made the posting.
<b>Request name</b>	chatNewPosting
<b>Request elements</b>	<p><b>String[] &lt;applds&gt;</b> : Contains at least one appld which has subscribed for this event.</p> <p><b>ChatPosting &lt;chatPosting&gt;</b> : Structure containing new chat text, creation date, etc..</p>
<b>Response elements</b>	(no parameters)

#### 9.3.7.2 chatNewPosting Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciEventInterface">EventLanguage</language>
    <user xmlns="urn:IciEventInterface">EventUser</user>
  </soap:Header>
  <soap:Body>
    
```



```
<chatNewPosting xmlns="urn:IciEventInterface">
  <appIds>
    <item>string</item>
    <item>string</item>
  </appIds>
  <chatPosting>
    <chatSessionId>string</chatSessionId>
    <postDate>string</postDate>
    <chatParticipant>string</chatParticipant>
    <contentText>string</contentText>
    <systemMessage>boolean</systemMessage>
  </chatPosting>
</chatNewPosting>
</soap:Body>
</soap:Envelope>
```

### 9.3.7.3 chatNewPostingResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <chatNewPostingResponse xmlns="urn:IciEventInterface">
      <response />
    </chatNewPostingResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.3.8 actionItemChanged Operation

### 9.3.8.1 Definition

<b>Purpose</b>	Signals a change of an Action Item and delivers a new Action Item Data.
<b>Request name</b>	actionItemChanged
<b>Request elements</b>	<p><b>String[] &lt;applds&gt;</b> : Contains at least one appld which has subscribed for this event.</p> <p><b>String[] &lt;userIds&gt;</b> : Not used.</p> <p><b>ActionSession &lt;actionSession&gt;</b> : Status for the Action Item.</p> <p><b>String &lt;actionItemId&gt;</b> : Id for the Action Item.</p> <p><b>ActionItem &lt;itemData&gt;</b> : Data of the Action Item.</p>



	<p><b>String &lt;scenarioId&gt;</b> : Not used (was <b>TextElement</b> Business Routing scenario ID).</p> <p>This operation has evolved between different CRM versions, and currently there are some unused request elements like userIds and scenarioId and some information is duplicated, like itemData containing a copy of actionSession status. Currently only actionItemId is used instead of actionSession.actionSessionId. It is unclear what information actually is used by CRM. itemData contains actionItemAttributes which originally come from iciActionItem_route operation. Some CRM versions use itemData.requestId and processId to identify the action item while CRM 7.0 does not seem to use it, but uses actionItemId instead. So the information presented here may not be correct.</p>
<p><b>Response elements</b></p>	<p>(no parameters)</p>

### 9.3.8.2 actionItemChanged Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciEventInterface">EventLanguage</language>
    <user xmlns="urn:IciEventInterface">EventUser</user>
  </soap:Header>
  <soap:Body>
    <actionItemChanged xmlns="urn:IciEventInterface">
      <appIds>
        <item>string</item>
        <item>string</item>
      </appIds>
      <userIds>
        <item>string</item>
        <item>string</item>
      </userIds>
      <actionSession>
        <actionParticipants>
          <item>string</item>
          <item>string</item>
        </actionParticipants>
        <capabilityList>
          <item>int</item>
          <item>int</item>
        </capabilityList>
        <title>string</title>
        <actionSessionId>string</actionSessionId>
        <actionStatus>
          <id>string</id>
          <description>string</description>
        </actionStatus>
      </actionItemChanged>
    </soap:Body>
  </soap:Envelope>
    
```



```

    <processingStatus>
      <id>string</id>
      <description>string</description>
    </processingStatus>
    <attachedData>string</attachedData>
    <actionLineId>string</actionLineId>
  </actionSession>
  <actionItemId>string</actionItemId>
  <itemData>
    <systemId>string</systemId>
    <client>string</client>
    <requestId>string</requestId>
    <processId>string</processId>
    <actionItemId>string</actionItemId>
    <actionItemInboxId>string</actionItemInboxId>
    <attachedData>string</attachedData>
    <capabilityList>
      <item>int</item>
      <item>int</item>
    </capabilityList>
    <processingStatus>
      <id>string</id>
      <description>string</description>
    </processingStatus>
    <actionItemStatus>
      <id>string</id>
      <description>string</description>
    </actionItemStatus>
    <actionItemAttributes>
      <item>
        <id>string</id>
        <description>string</description>
      </item>
      <item>
        <id>string</id>
        <description>string</description>
      </item>
    </actionItemAttributes>
  </itemData>
  <scenarioId>string</scenarioId>
</actionItemChanged>
</soap:Body>
</soap:Envelope>

```

### 9.3.8.3 actionItemChangedResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <actionItemChangedResponse xmlns="urn:IciEventInterface">
      <response />
    </actionItemChangedResponse>
  </soap:Body>
</soap:Envelope>

```





### 9.3.8.4 Sample Data

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <evnt:actionItemChanged xmlns:evnt="urn:IciEventInterface">
      <appIds>
        <item>479E5A480CD25382E1000000A4241B7</item>
      </appIds>
      <userIds />
      <actionSession>
        <actionParticipants>
          <item>GENTEST2</item>
        </actionParticipants>
        <capabilityList>
          <item>1</item>
          <item>2</item>
          <item>6</item>
          <item>7</item>
          <item>8</item>
          <item>403</item>
        </capabilityList>
        <title />
        <actionSessionId />
        <actionStatus>
          <description>Alerting</description>
          <id>3</id>
        </actionStatus>
        <processingStatus>
          <description>Active</description>
          <id>2</id>
        </processingStatus>
        <attachedData><?xml version="1.0" encoding="iso-8859-1"?>
          <ItemAttachedData>
            <Application
id="RRequestedSkills"></Application>
            <Application id="actionItemAttributes">
              <TO_ADDRESS>ERMS@505.Q6D.R3.SAP-
AG.DE</TO_ADDRESS>
              <TO_NAME>ERMS@505.Q6D.R3.SAP-
AG.DE</TO_NAME>
              <SENSITIVITY></SENSITIVITY>
              <SEND_DATE>20080128234007 </SEND_DATE>
              <SUBJECT>Test Octavian</SUBJECT>
            </Application>
          </ItemAttachedData>
        </attachedData>
        <FROM_ADDRESS>leo.flores@SAP.COM</FROM_ADDRESS>
        <SUBJECT_CHARSET></SUBJECT_CHARSET>
        <FROM_NAME>leo.flores@SAP.COM</FROM_NAME>
      </actionSession>
    </evnt:actionItemChanged>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```



```

<IMPORTANCE></IMPORTANCE>

<MAIL_ID>96F5ABFF45530146B419CA55363F429C06FDFCA1@uspale20.pal.sap.corp</MAIL_ID>
  </Application>
  <Application id="RTargetUsed">
    <TargetName>SAP agents</TargetName>
    <TargetType>2</TargetType>
  </Application>
  <Application id="GENESYS-CAD">

</processId>FOL29000000000004RAW33000000000128</processId>

<RTargetPlaceSelected>plsim_50002</RTargetPlaceSelected>
  <requestId>000001157144</requestId>
  <GPMC_Generated realName="PegAGSAP
agents" TKVType="TKVTypeInt">3</GPMC_Generated>
  <scenarioId></scenarioId>
  <RTenant>Resources</RTenant>

<RTargetTypeSelected>2</RTargetTypeSelected>
  <CustomNumber2
TKVType="TKVTypeInt">0</CustomNumber2>
  <RTargetObjectSelected>SAP
agents</RTargetObjectSelected>

<RRequestedSkillCombination></RRequestedSkillCombination>

<RTargetRuleSelected></RTargetRuleSelected>
  <RTargetAgentGroup>SAP
agents</RTargetAgentGroup>
  <RStrategyName>Action Item
routing</RStrategyName>
  <CustomNumber1
TKVType="TKVTypeInt">0</CustomNumber1>
  <CustomNumber3
TKVType="TKVTypeInt">0</CustomNumber3>
  <PegRejected
TKVType="TKVTypeInt">256</PegRejected>

<RTargetAgentSelected>SAP_AGENT_2</RTargetAgentSelected>
  <gplus-analytics-
id>00076a3SV6ND0013</gplus-analytics-id>
  </Application>
  </ItemAttachedData>
  </attachedData>
  <actionLineId />
</actionSession>
  <actionItemId>00076a3SV6ND0013</actionItemId>
  <itemData>
  <systemId />

```



```

        <client />
        <requestId>000001157144</requestId>
        <processId>FOL29000000000004RAW33000000000128</processId>
        <attachedData><?xml version="1.0" encoding="iso-8859-1"?>
            <ItemAttachedData>
                <Application
id="RRequestedSkills"></Application>
                <Application id="actionItemAttributes">
                    <TO_ADDRESS>ERMS@505.Q6D.R3.SAP-
AG.DE</TO_ADDRESS>
                    <TO_NAME>ERMS@505.Q6D.R3.SAP-
AG.DE</TO_NAME>
                    <SENSITIVITY></SENSITIVITY>
                    <SEND_DATE>20080128234007 </SEND_DATE>
                    <SUBJECT>Test Octavian</SUBJECT>
                    <FROM_ADDRESS>leo.flores@SAP.COM</FROM_ADDRESS>
                    <SUBJECT_CHARSET></SUBJECT_CHARSET>
                    <FROM_NAME>leo.flores@SAP.COM</FROM_NAME>
                    <IMPORTANCE></IMPORTANCE>
                    <MAIL_ID>96F5ABFF45530146B419CA55363F429C06FDFCA1@uspale20.pal.sap.corp</MAIL_ID>
                    </Application>
                    <Application id="RTargetUsed">
                        <TargetName>SAP agents</TargetName>
                        <TargetType>2</TargetType>
                    </Application>
                    <Application id="GENESYS-CAD">
                        <processId>FOL29000000000004RAW33000000000128</processId>
                        <RTargetPlaceSelected>plsim_50002</RTargetPlaceSelected>
                        <requestId>000001157144</requestId>
                        <GPMC_Generated realName="PegAGSAP
agents" TKVType="TKVTypeInt">3</GPMC_Generated>
                        <scenarioId></scenarioId>
                        <RTenant>Resources</RTenant>
                        <RTargetTypeSelected>2</RTargetTypeSelected>
                        <CustomNumber2
TKVType="TKVTypeInt">0</CustomNumber2>
                        <RTargetObjectSelected>SAP
agents</RTargetObjectSelected>
                        <RRequestedSkillCombination></RRequestedSkillCombination>
                        <RTargetRuleSelected></RTargetRuleSelected>
                        <RTargetAgentGroup>SAP
agents</RTargetAgentGroup>

```



```

                                <RstrategyName>Action Item
routing</RstrategyName>

                                <CustomNumber1
TKVType="TKVTypeInt">0</CustomNumber1>

                                <CustomNumber3
TKVType="TKVTypeInt">0</CustomNumber3>

                                <PegRejected
TKVType="TKVTypeInt">256</PegRejected>

                                <RTargetAgentSelected>SAP_AGENT_2</RTargetAgentSelected>
                                <gplus-analytics-
id>00076a3SV6ND0013</gplus-analytics-id>

                                </Application>
                                </ItemAttachedData>
                                </attachedData>
                                <capabilityList>
                                    <item>1</item>
                                    <item>2</item>
                                    <item>6</item>
                                    <item>7</item>
                                    <item>8</item>
                                    <item>403</item>
                                </capabilityList>
                                <processingStatus>
                                    <description>Active</description>
                                    <id>2</id>
                                </processingStatus>
                                <actionItemStatus>
                                    <description>Alerting</description>
                                    <id>3</id>
                                </actionItemStatus>
                                <actionItemAttributes>
                                    <item>
                                        <id>TO_ADDRESS</id>
                                        <description>ERMS@505.Q6D.R3.SAP-
AG.DE</description>
                                    </item>
                                    <item>
                                        <id>SENSITIVITY</id>
                                        <description />
                                    </item>
                                    <item>
                                        <id>TO_NAME</id>
                                        <description>ERMS@505.Q6D.R3.SAP-
AG.DE</description>
                                    </item>
                                    <item>
                                        <id>SUBJECT</id>
                                        <description>Test Octavian</description>
                                    </item>

```



```

        <item>
            <id>SEND_DATE</id>
            <description>20080128234007</description>
        </item>
        <item>
            <id>FROM_ADDRESS</id>
            <description>leo.flores@SAP.COM</description>
        </item>
        <item>
            <id>SUBJECT_CHARSET</id>
            <description />
        </item>
        <item>
            <id>IMPORTANCE</id>
            <description />
        </item>
        <item>
            <id>FROM_NAME</id>
            <description>leo.flores@SAP.COM</description>
        </item>
        <item>
            <id>MAIL_ID</id>
            <description>96F5ABFF45530146B419CA55363F429C06FDCA1@uspale20.pal.sap.corp</descripti
on>
        </item>
    </actionItemAttributes>
</itemData>
<scenarioId />
</evnt:actionItemChanged>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 9.4 Iciltem Interface

### 9.4.1 create Operation

#### 9.4.1.1 Definition

<b>Purpose</b>	The application creates a contact center item with the CREATED status.
<b>Request name</b>	Create



<p><b>Request elements</b></p>	<p><b>String &lt;channelType&gt;</b> : The channel type:</p> <ul style="list-style-type: none"> <li>1 = Telephony (phone call)</li> <li>2 = Messaging (e-mail, fax or pager/SMS message)</li> <li>3 = Chat session</li> <li>4 = Action routing</li> </ul> <p><b>String &lt;containerId&gt;</b> : Matches the type:</p> <ul style="list-style-type: none"> <li>1 : Phone line number</li> <li>2 : Folder name</li> <li>3 : Chat line name</li> <li>4 : Action item inbox</li> </ul>
<p><b>Response elements</b></p>	<p><b>Item &lt;response&gt;</b> : See the Item element for more information.</p>

### 9.4.1.2 create Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <create xmlns="urn:IciItemInterface">
      <channelType>1</channelType>
      <containerId>+129</containerId>
    </create>
  </soap:Body>
</soap:Envelope>
```

### 9.4.1.3 createResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns="urn:IciItemInterface">
      <response>
      <status>
```



```

        <id>1</id>
        <description>created</description>
    </status>
    <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
    <processingStatus>
        <id>1</id>
        <description>not in process</description>
    </processingStatus>
    <capabilityList>
        <item>6</item>
        <item>7</item>
        <item>8</item>
        <item>101</item>
        <item>102</item>
        <item>103</item>
    </capabilityList>
    </response>
</createResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.4.2 wrapUpRequired Operation

### 9.4.2.1 Definition

<p><b>Purpose</b></p>	<p>The operation informs the contact center that the wrap-up work is required for this item, and that the item status changes to “Wrap up” as soon as the current processing of the item is finished. This is indicated by an appropriate SOAP method (for example, IciPhoneCall_drop).</p> <p>The application uses this method only if the user who works on the item is in the “requested wrap up status” wrap-up mode. If the user is in the “automatic wrap up status” wrap-up mode, the contact center must always set the item status to “Wrap up” without having received this SOAP message.</p>
<p><b>Request name</b></p>	<p>wrapUpRequired</p>
<p><b>Request elements</b></p>	<p><b>String &lt;itemId&gt;</b> : The ID of the item.</p>
<p><b>Response elements</b></p>	<p>(no parameters)</p>

### 9.4.2.2 wrapUpRequired Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    
```



```

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <wrapUpRequired xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
    </wrapUpRequired>
  </soap:Body>
</soap:Envelope>
    
```

### 9.4.2.3 wrapUpRequiredResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <wrapUpRequiredResponse xmlns="urn:IciItemInterface">
      <response />
    </wrapUpRequiredResponse>
  </soap:Body>
</soap:Envelope>
    
```

## 9.4.3 wrapUpEnded Operation

### 9.4.3.1 Definition

<b>Purpose</b>	The operation informs the contact center that the wrap up work for this item has been finished, and that the item status now changes to “Not in process”.
<b>Request name</b>	wrapUpEnded
<b>Request elements</b>	<b>String &lt;itemId&gt;</b> : The ID of the item.
<b>Response elements</b>	(no parameters)

### 9.4.3.2 wrapUpEnded Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    
```





```

    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
  <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
</soap:Header>
<soap:Body>
  <wrapUpEnded xmlns="urn:IciItemInterface">
    <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
  </wrapUpEnded>
</soap:Body>
</soap:Envelope>

```

### 9.4.3.3 wrapUpEndedResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <wrapUpEndedResponse xmlns="urn:IciItemInterface">
      <response />
    </wrapUpEndedResponse>
  </soap:Body>
</soap:Envelope>

```

## 9.4.4 accept operation

### 9.4.4.1 Definition

<b>Purpose</b>	<p>The operation informs that a user accepts an assigned item. The processing status changes to “active”. In addition to that, the channel-specific functionality is triggered:</p> <ol style="list-style-type: none"> <li>1. Telephony : The call will be connected.</li> <li>2. Messaging : The message may be marked as "read", and a notification may be sent to the original sender.</li> <li>3. Chat : The new chat participant is reported to the other participants.</li> <li>4. Action Routing : N/A</li> </ol>
<b>Request name</b>	Accept
<b>Request elements</b>	<p><b>String &lt;itemId&gt;</b> : The ID of the item.</p> <p><b>String &lt;containerId&gt;</b> : The container that contains the item.</p>



<b>Response elements</b>	(no parameters)
--------------------------	-----------------

### 9.4.4.2 accept Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <accept xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
      <containerId>+129</containerId>
    </accept>
  </soap:Body>
</soap:Envelope>
```

### 9.4.4.3 acceptResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <acceptResponse xmlns="urn:IciItemInterface">
      <response />
    </acceptResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.4.5 reroute Operation

### 9.4.5.1 Definition

<b>Purpose</b>	The operation informs that a user does not accept an assigned item. The item must be rerouted by the contact center without routing it to the current user again.
<b>Request name</b>	Reroute
<b>Request elements</b>	<b>String &lt;itemId&gt;</b> : The ID of the item. <b>String &lt;containerId&gt;</b> : The container that contains the item.



<b>Response elements</b>	(no parameters)
--------------------------	-----------------

### 9.4.5.2 reroute Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <reroute xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
      <containerId>+129</containerId>
    </reroute>
  </soap:Body>
</soap:Envelope>
```

### 9.4.5.3 rerouteResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <rerouteResponse xmlns="urn:IciItemInterface">
      <response />
    </rerouteResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.4.6 forward Operation

### 9.4.6.1 Definition

<b>Purpose</b>	The operation informs that a user forwards an item to another user (identified by the container) or to a queue.
<b>Request name</b>	Forward
<b>Request elements</b>	<p><b>String &lt;itemId&gt;</b> : The ID of the item.</p> <p><b>String &lt;containerId&gt;</b> : The container from which the item must be transferred.</p>



	<p><b>String &lt;destType&gt;</b> : The type of the transfer destination. One of the following values:</p> <ul style="list-style-type: none"> <li>1 = Phone line</li> <li>2 = Folder</li> <li>3 = Chat line</li> <li>4 = Action item inbox</li> <li>12 = Queue</li> </ul> <p><b>String &lt;destination&gt;</b> : The container ID or the queue ID.</p>
<b>Response elements</b>	(no parameters)

### 9.4.6.2 forward Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <forward xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
      <containerId>+129</containerId>
      <destType>1</destType>
      <destination>+130</destination>
    </forward>
  </soap:Body>
</soap:Envelope>
```

### 9.4.6.3 forwardResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <forwardResponse xmlns="urn:IciItemInterface">
      <response />
    </forwardResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.4.7 getAttachedData Operation

### 9.4.7.1 Definition

<b>Purpose</b>	The operation reads data attached to an item (for example, data related to a phone call).
<b>Request name</b>	getAttachedData
<b>Request elements</b>	<b>String &lt;itemId&gt;</b> : The ID of the item.
<b>Response elements</b>	<b>String &lt;attachedData&gt;</b> : An XML document containing attached data. If there is no data, an empty string (not a SOAP fault message) is sent back.

### 9.4.7.2 getAttachedData Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <getAttachedData xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
    </getAttachedData>
  </soap:Body>
</soap:Envelope>
```

### 9.4.7.3 getAttachedDataResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttachedDataResponse xmlns="urn:IciItemInterface">
      <response>XML</response>
    </getAttachedDataResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.4.8 setAttachedData Operation

### 9.4.8.1 Definition

<b>Purpose</b>	The operation sets the attached data for an item.
<b>Request name</b>	setAttachedData
<b>Request elements</b>	<p><b>String &lt;itemId&gt;</b> : The ID of the item.</p> <p><b>String &lt;attachedData&gt;</b> : An XML document containing attached data. May be empty.</p>
<b>Response elements</b>	(no parameters)

### 9.4.8.2 setAttachedData Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <setAttachedData xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
      <attachedData>XML</attachedData>
    </setAttachedData>
  </soap:Body>
</soap:Envelope>
```

### 9.4.8.3 setAttachedDataResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setAttachedDataResponse xmlns="urn:IciItemInterface">
      <response />
    </setAttachedDataResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.4.9 deleteAttachedData Operation

### 9.4.9.1 Definition

<b>Purpose</b>	The operation deletes the data attached to an item.
<b>Request name</b>	deleteAttachedData
<b>Request elements</b>	String <itemId> : The ID of the item.
<b>Response elements</b>	(no parameters)

### 9.4.9.2 deleteAttachedData Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciItem/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <deleteAttachedData xmlns="urn:IciItemInterface">
      <itemId>A163FDACEE7A476BAB59E15C81B6D4F2</itemId>
    </deleteAttachedData>
  </soap:Body>
</soap:Envelope>
```

### 9.4.9.3 deleteAttachedDataResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <deleteAttachedDataResponse xmlns="urn:IciItemInterface">
      <response />
    </deleteAttachedDataResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.5 IciContainer Interface

### 9.5.1 subscribe Operation

#### 9.5.1.1 Definition

<b>Purpose</b>	If the application subscribes to a container, the container events as well as item events for that container are expected. The subscription key consists of appUrl and appld, i.e. the application may send multiple subscriptions with the same appUrl but different applds. In that case the contact center may collect events of the same type and content.
<b>Request name</b>	Subscribe
<b>Request elements</b>	<p><b>String &lt;appUrl&gt;</b> : Identifies the application and represents the address of the SOAP entry point to which the events have to be sent to. The format follows the URL rules, for example http://sapserver:80/soapdispatcher.</p> <p><b>String &lt;appld&gt;</b> : Identifies the subscribing application instance.</p> <p><b>String &lt;channelType&gt;</b> : Specifies the type of events:</p> <p>1 = containerChanged (Phone line) + phoneCallChanged</p> <p>2 = containerChanged (Folder) + messageChanged</p> <p>3 = containerChanged (Chat line) + chatChanged</p> <p>4 = containerChanged (Action item inbox) + actionItemChanged</p> <p><b>String &lt;containerId&gt;</b> : The Container ID (for example, the phone number of the phone line or the folder name).</p>
<b>Response elements</b>	(no parameters)

#### 9.5.1.2 subscribe Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```





```

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciContainer/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciContainer/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <subscribe xmlns="urn:IciContainerInterface">

<appUrl>http://ip:50000/IciEventService/IciEventConf?style=document</appUrl>
    <appId>1cb65470-a2c1-11da-aa97-001143dc2e6e</appId>
    <channelType>1</channelType>
    <containerId>+129</containerId>
    </subscribe>
  </soap:Body>
</soap:Envelope>
    
```

### 9.5.1.3 subscribeResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <subscribeResponse xmlns="urn:IciContainerInterface">
      <response />
    </subscribeResponse>
  </soap:Body>
</soap:Envelope>
    
```

## 9.5.2 unsubscribe Operation

### 9.5.2.1 Definition

<b>Purpose</b>	The application must call an unsubscription if it is not interested in receiving events anymore.
<b>Request name</b>	Unsubscribe
<b>Request elements</b>	<b>String &lt;appUrl&gt;</b> : Identifies the application and represents the address of the SOAP entry point to which the events have to be sent to. The format follows the URL rules, for example http://sapserver:80/soapdispatcher.



	<b>String &lt;appld&gt;</b> : Identifies the subscribing application instance.
<b>Response elements</b>	(no parameters)

### 9.5.2.2 unsubscribe Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciContainer/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciContainer/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <unsubscribe xmlns="urn:IciContainerInterface">

<appUrl>http://ip:50000/IciEventService/IciEventConf?style=document</appUrl>
    <appId>1cb65470-a2c1-11da-aa97-001143dc2e6e</appId>
    </unsubscribe>
  </soap:Body>
</soap:Envelope>
```

### 9.5.2.3 unsubscribeResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <unsubscribeResponse xmlns="urn:IciContainerInterface">
      <response />
    </unsubscribeResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.6 IciPhoneLine Interface

### 9.6.1 getCalls Operation

#### 9.6.1.1 Definition

<b>Purpose</b>	The operation gets all existing phone calls with the phone line data.
<b>Request name</b>	getCalls
<b>Request elements</b>	<b>String &lt;lineNumber&gt; : = containerId</b> : Get all calls of this phone number.
<b>Response elements</b>	<b>PhoneCall[] &lt;phoneCalls&gt;</b> : An array of calls.

#### 9.6.1.2 getCalls Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneLine/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneLine/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <getCalls xmlns="urn:IciPhoneLineInterface">
      <lineNumber>+129</lineNumber>
    </getCalls>
  </soap:Body>
</soap:Envelope>
```

#### 9.6.1.3 getCallsResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getCallsResponse xmlns="urn:IciPhoneLineInterface">
      <response>
        <item>
          <lineNumber>string</lineNumber>
          <callHandle>string</callHandle>
        </item>
      </response>
    </getCallsResponse>
  </soap:Body>
</soap:Envelope>
```



```

        <processingStatus>
            <id>string</id>
            <description>string</description>
        </processingStatus>
        <remoteNumbers>
            <item>string</item>
            <item>string</item>
        </remoteNumbers>
        <dialedNumber>string</dialedNumber>
        <callStatus>
            <id>string</id>
            <description>string</description>
        </callStatus>
        <capabilityList>
            <item>int</item>
            <item>int</item>
        </capabilityList>
        <inbound>boolean</inbound>
        <internal>boolean</internal>
        <attachedData>string</attachedData>
        <trunkId>string</trunkId>
    </item>
</response>
</getCallsResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.6.2 dropCalls Operation

### 9.6.2.1 Definition

<b>Purpose</b>	The operation disconnects all calls of an extension number.
<b>Request name</b>	dropCalls
<b>Request elements</b>	<b>String &lt;lineNumber&gt;</b> : Identifies the phone number.
<b>Response elements</b>	(no parameters)

### 9.6.2.2 dropCalls Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneLine/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneLine/literalTypes">string</user>
  </soap:Header>
    
```



```
<soap:Body>
  <dropCalls xmlns="urn:IciPhoneLineInterface">
    <lineNumber>+129</lineNumber>
  </dropCalls>
</soap:Body>
</soap:Envelope>
```

### 9.6.2.3 dropCallsResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dropCallsResponse xmlns="urn:IciPhoneLineInterface">
      <response />
    </dropCallsResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.7 IciPhoneCall Interface

### 9.7.1 getAttributes Operation

#### 9.7.1.1 Definition

<b>Purpose</b>	The operation gets the attributes of a specific phone call.
<b>Request name</b>	getAttributes
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line that contains the call.</p> <p><b>String &lt;callHandle&gt; : = itemId</b></p>
<b>Response elements</b>	<b>PhoneCall &lt;phoneCall&gt;</b> : The call information.

#### 9.7.1.2 getAttributes Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
```



```

xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
  <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
</soap:Header>
<soap:Body>
  <getAttributes xmlns="urn:IciPhoneCallInterface">
    <lineNumber>+129</lineNumber>
    <callHandle>A163FDACEE7A476BAB59E15C81B6D4F2</callHandle>
  </getAttributes>
</soap:Body>
</soap:Envelope>
    
```

### 9.7.1.3 getAttributesResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttributesResponse xmlns="urn:IciPhoneCallInterface">
      <response>
        <lineNumber>string</lineNumber>
        <callHandle>string</callHandle>
        <processingStatus>
          <id>string</id>
          <description>string</description>
        </processingStatus>
        <remoteNumbers>
          <item>string</item>
          <item>string</item>
        </remoteNumbers>
        <dialNumber>string</dialNumber>
        <callStatus>
          <id>string</id>
          <description>string</description>
        </callStatus>
        <capabilityList>
          <item>int</item>
          <item>int</item>
        </capabilityList>
        <inbound>boolean</inbound>
        <internal>boolean</internal>
        <attachedData>string</attachedData>
        <trunkId>string</trunkId>
      </response>
    </getAttributesResponse>
  </soap:Body>
</soap:Envelope>
    
```

## 9.7.2 dial Operation

### 9.7.2.1 Definition

<b>Purpose</b>	The operation creates a new outbound phone call. If the application creates the phone call, it has already called
----------------	-------------------------------------------------------------------------------------------------------------------



	Icitem_create before to get a call handle (itemId) of a new phone call item.
<b>Request name</b>	Dial
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;callHandle&gt; : = itemId</b> : The call handle of a call which must be dialed.</p> <p><b>String &lt;dialNumber&gt;</b> : The number to dial.</p>
<b>Response elements</b>	<b>String &lt;dialedNumber&gt;</b> : The number that was dialed (may be different from the input parameter).

### 9.7.2.2 dial Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <dial xmlns="urn:IciPhoneCallInterface">
      <lineNumber>+129</lineNumber>
      <callHandle>A163FDACEE7A476BAB59E15C81B6D4F2</callHandle>
      <dialNumber>+130</dialNumber>
    </dial>
  </soap:Body>
</soap:Envelope>
```

### 9.7.2.3 dialResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dialResponse xmlns="urn:IciPhoneCallInterface">
      <response>+130</response>
    </dialResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.7.3 consult Operation

### 9.7.3.1 Definition

<b>Purpose</b>	When the application must create a consultation call, first it invokes <code>IciItem.create</code> (like for an outbound call) and receives a <code>callHandle</code> . Then it calls <code>IciPhoneCall.consult</code> with that specific call handle plus the call handle of the related connected call.
<b>Request name</b>	Consult
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;callHandle&gt; : = itemId</b> : The call with the “created” status which becomes the consultation call.</p> <p><b>String &lt;connCallHandle&gt; : = itemId</b> : The connected call to be put on hold and which probably will be transferred to or diverted with the new call.</p> <p><b>String &lt;dialNumber&gt;</b> : The number to dial.</p> <p><b>Int &lt;nextStep&gt;</b> : One of the following values:</p> <ul style="list-style-type: none"> <li>1 = unknown</li> <li>2 = transfer</li> <li>3 = conference</li> <li>4 = reconnect</li> </ul>
<b>Response elements</b>	(no parameters)

### 9.7.3.2 consult Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
```





```

    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
</soap:Header>
<soap:Body>
  <consult xmlns="urn:IciPhoneCallInterface">
    <lineNumber>string</lineNumber>
    <callHandle>string</callHandle>
    <connCallHandle>string</connCallHandle>
    <dialNumber>string</dialNumber>
    <nextStep>int</nextStep>
  </consult>
</soap:Body>
</soap:Envelope>

```

### 9.7.3.3 consultResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <consultResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </consultResponse>
  </soap:Body>
</soap:Envelope>

```

## 9.7.4 transfer Operation

### 9.7.4.1 Definition

<b>Purpose</b>	The operation transfers a connected call to another extension number. Does not wait.
<b>Request name</b>	Transfer
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;connCallHandle&gt; : = itemId</b> : The connected consultation call to the target number.</p> <p><b>String &lt;heldCallHandle&gt; : = itemId</b> : The held call to be transferred.</p>
<b>Response elements</b>	(no parameters)



### 9.7.4.2 transfer Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <transfer xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <connCallHandle>string</connCallHandle>
      <heldCallHandle>string</heldCallHandle>
    </transfer>
  </soap:Body>
</soap:Envelope>
```

### 9.7.4.3 transferResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <transferResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </transferResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.7.5 reconnect Operation

### 9.7.5.1 Definition

<b>Purpose</b>	The operation disconnects a connected call and releases a specific held call.
<b>Request name</b>	Reconnect
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;connCallHandle&gt; : = itemId</b> : The connected call to disconnect.</p>



	<b>String &lt;heldCallHandle&gt; : = itemId</b> : The held call to release.
<b>Response elements</b>	(no parameters)

### 9.7.5.2 reconnect Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <reconnect xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <connCallHandle>string</connCallHandle>
      <heldCallHandle>string</heldCallHandle>
    </reconnect>
  </soap:Body>
</soap:Envelope>
```

### 9.7.5.3 reconnectResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <reconnectResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </reconnectResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.7.6 drop Operation

### 9.7.6.1 Definition

<b>Purpose</b>	The operation disconnects a call.
<b>Request name</b>	Drop



<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;callHandle&gt; : = itemId</b> : The call to disconnect.</p>
<b>Response elements</b>	(no parameters)

### 9.7.6.2 drop message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <drop xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <callHandle>string</callHandle>
    </drop>
  </soap:Body>
</soap:Envelope>
```

### 9.7.6.3 dropResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dropResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </dropResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.7.7 hold Operation

### 9.7.7.1 Definition

<b>Purpose</b>	The operation puts a connected call on hold.
<b>Request name</b>	Hold



<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;callHandle&gt; : = itemId</b> : The call to put on hold.</p>
<b>Response elements</b>	(no parameters)

### 9.7.7.2 hold Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <hold xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <callHandle>string</callHandle>
    </hold>
  </soap:Body>
</soap:Envelope>
```

### 9.7.7.3 holdResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <holdResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </holdResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.7.8 unhold Operation

### 9.7.8.1 Definition

<b>Purpose</b>	The operation releases a held call.
<b>Request name</b>	Unhold



<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;callHandle&gt; : = itemId</b> : The call to release.</p>
<b>Response elements</b>	(no parameters)

### 9.7.8.2 unhold Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <unhold xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <callHandle>string</callHandle>
    </unhold>
  </soap:Body>
</soap:Envelope>
```

### 9.7.8.3 unholdResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <unholdResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </unholdResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.7.9 alternate Operation

### 9.7.9.1 Definition

<b>Purpose</b>	The operation toggles between a connected and a held call.
<b>Request name</b>	Alternate



<b>Request elements</b>	<p>String &lt;lineNumber&gt; := containerId : The phone line to use.</p> <p>String &lt;connCallHandle&gt; := itemId : The connected call to put on hold.</p> <p>String &lt;heldCallHandle&gt; := itemId : The held call to release.</p>
<b>Response elements</b>	(no parameters)

### 9.7.9.2 alternate Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <alternate xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <connCallHandle>string</connCallHandle>
      <heldCallHandle>string</heldCallHandle>
    </alternate>
  </soap:Body>
</soap:Envelope>
```

### 9.7.9.3 alternateResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <alternateResponse xmlns="urn:IciPhoneCallInterface">
      <response />
    </alternateResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.7.10 conference Operation

### 9.7.10.1 Definition

<b>Purpose</b>	Creates a phone conference by connecting a connected call and a held call. One or both of them may be a conference call already. It is not specified which call is dropped and which call takes over the remoteNumbers of the dropped call. That information must be propagated by events.
<b>Request name</b>	conference
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;connCallHandle&gt; : = itemId</b> : The connected call to add to conference.</p> <p><b>String &lt;heldCallHandle&gt; : = itemId</b> : The held call to add to conference.</p>
<b>Response elements</b>	(no parameters)

### 9.7.10.2 conference Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <conference xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <connCallHandle>string</connCallHandle>
      <heldCallHandle>string</heldCallHandle>
    </conference>
  </soap:Body>
</soap:Envelope>
```

### 9.7.10.3 conferenceResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```





```

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <conferenceResponse xmlns="urn:IciPhoneCallInterface">
      <response xmlns="" />
    </conferenceResponse>
  </soap:Body>
</soap:Envelope>
    
```

## 9.7.11 dropParty Operation

### 9.7.11.1 Definition

<b>Purpose</b>	Drops another extension from an existing conference call.
<b>Request name</b>	dropParty
<b>Request elements</b>	<p><b>String &lt;lineNumber&gt; : = containerId</b> : The phone line to use.</p> <p><b>String &lt;remoteNumber&gt;</b> : Extension to drop.</p> <p><b>String &lt;callHandle&gt; : = itemId</b> : Identifies conference call.</p>
<b>Response elements</b>	(no parameters)

### 9.7.11.2 dropParty Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciPhoneCall/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <dropParty xmlns="urn:IciPhoneCallInterface">
      <lineNumber>string</lineNumber>
      <remoteNumber>string</remoteNumber>
      <callHandle>string</callHandle>
    </dropParty>
  </soap:Body>
</soap:Envelope>
    
```



### 9.7.11.3 dropPartyResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <dropPartyResponse xmlns="urn:IciPhoneCallInterface">
      <response xmlns="" />
    </dropPartyResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.8 IciFolder Interface

### 9.8.1 getMessages Operation

#### 9.8.1.1 Definition

<b>Purpose</b>	The operation gets the list of messages that are in one folder.
<b>Request name</b>	getMessages
<b>Request elements</b>	<b>String &lt;folder&gt; : = containerId</b>
<b>Response elements</b>	<b>Message[]&lt;messages&gt; : The list of messages.</b>

#### 9.8.1.2 getMessages Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciFolder/literalTypes">string1</language>
    <user xmlns="http://wicom.com/ws/OII/IciFolder/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <getMessages xmlns="urn:IciFolderInterface">
      <folder>string</folder>
    </getMessages>
  </soap:Body>
</soap:Envelope>
```



### 9.8.1.3 getMessagesResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getMessagesResponse xmlns="urn:IciFolderInterface">
      <response>
        <item>
          <folder>string</folder>
          <subject>string</subject>
          <messageId>string</messageId>
          <processingStatus>
            <id>string</id>
            <description>string</description>
          </processingStatus>
          <originator>string</originator>
          <messageText>string</messageText>
          <attachments>
            <item>string</item>
            <item>string</item>
          </attachments>
          <messageStatus>
            <id>string</id>
            <description>string</description>
          </messageStatus>
          <capabilityList>
            <item>int</item>
            <item>int</item>
          </capabilityList>
          <attachedData>string</attachedData>
          <originalMessageId>string</originalMessageId>
          <notification>boolean</notification>
          <read>boolean</read>
          <answered>boolean</answered>
          <forwarded>boolean</forwarded>
        </item>
      </response>
    </getMessagesResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.9 IciMessage Interface

### 9.9.1 getAttributes Operation

#### 9.9.1.1 Definition

<b>Purpose</b>	The operation gets the most important message attributes (sender, subject, main body text, attachment list, etc.), i.e. it returns almost the entire message excluding the contents of the attachments.
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<b>Request name</b>	getAttributes
<b>Request elements</b>	<b>String &lt;messageId&gt; : = itemId</b>
<b>Response elements</b>	<b>Message &lt;message&gt; : The message information.</b>

### 9.9.1.2 getAttributes Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <getAttributes xmlns="urn:IciMessageInterface">
      <messageId>string</messageId>
    </getAttributes>
  </soap:Body>
</soap:Envelope>
```

### 9.9.1.3 getAttributesResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttributesResponse xmlns="urn:IciMessageInterface">
      <response>
        <folder>string</folder>
        <subject>string</subject>
        <messageId>string</messageId>
        <processingStatus>
          <id>string</id>
          <description>string</description>
        </processingStatus>
        <originator>string</originator>
        <messageText>string</messageText>
        <attachments>
          <item>string</item>
          <item>string</item>
        </attachments>
        <messageStatus>
          <id>string</id>
          <description>string</description>
        </messageStatus>
        <capabilityList>
          <item>int</item>
          <item>int</item>
        </capabilityList>
      </response>
    </getAttributesResponse>
  </soap:Body>
</soap:Envelope>
```



```

        </capabilityList>
        <attachedData>string</attachedData>
        <originalMessageId>string</originalMessageId>
        <notification>boolean</notification>
        <read>boolean</read>
        <answered>boolean</answered>
        <forwarded>boolean</forwarded>
    </response>
</getAttributesResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.9.2 getContent Operation

### 9.9.2.1 Definition

<b>Purpose</b>	The operation gets the complete native contents of a message (header + body) as the RFC 822 / MIME representation.
<b>Request name</b>	getContent
<b>Request elements</b>	<b>String &lt;messageId&gt; : = itemId</b>
<b>Response elements</b>	<b>String &lt;content&gt; : The complete RFC 822 compliant contents of the message.</b>

### 9.9.2.2 getContent Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <getContent xmlns="urn:IciMessageInterface">
      <messageId>string</messageId>
    </getContent>
  </soap:Body>
</soap:Envelope>
    
```



### 9.9.2.3 getContentResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getContentResponse xmlns="urn:IciMessageInterface">
      <response>string</response>
    </getContentResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.9.3 setContent Operation

### 9.9.3.1 Definition

<b>Purpose</b>	The operation sets the complete native contents of a message (header + body) as the RFC 822 / MIME presentation.
<b>Request name</b>	setContent
<b>Request elements</b>	<b>String &lt;messageId&gt; : = itemId</b> <b>String &lt;content&gt; : The complete RFC 822 compliant contents of the message.</b>
<b>Response elements</b>	(no parameters)

### 9.9.3.2 setContent Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <setContent xmlns="urn:IciMessageInterface">
      <messageId>string</messageId>
      <content>string</content>
    </setContent>
  </soap:Body>
</soap:Envelope>
```



### 9.9.3.3 setContentResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setContentResponse xmlns="urn:IciMessageInterface">
      <response />
    </setContentResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.9.4 delete Operation

### 9.9.4.1 Definition

<b>Purpose</b>	The operation deletes an existing message.
<b>Request name</b>	Delete
<b>Request elements</b>	<b>String &lt;messageId&gt; : = itemId</b>
<b>Response elements</b>	(no parameters)

### 9.9.4.2 delete Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string1</language>
    <user
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string</user>
  </soap:Header>
  <soap:Body>
    <delete xmlns="urn:IciMessageInterface">
      <messageId>string</messageId>
    </delete>
  </soap:Body>
</soap:Envelope>
```



### 9.9.4.3 deleteResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <deleteResponse xmlns="urn:IciMessageInterface">
      <response />
    </deleteResponse>
  </soap:Body>
</soap:Envelope>
```

### 9.9.5 send Operation

#### 9.9.5.1 Definition

<b>Purpose</b>	The operation sends the message to the recipients specified in the e-mail header. The Bcc (blind carbon copy) recipients will also be specified in the header with a “Bcc:” line which has to be removed at the contact center or its associated mail server.
<b>Request name</b>	Send
<b>Request elements</b>	<p><b>String &lt;messageId&gt; : = itemId</b></p> <p><b>Boolean &lt;requestSuccessNotification&gt; :</b> The contact center must request a delivery notification when sending the message.</p> <p><b>Boolean &lt;requestFailureNotification&gt; :</b> The contact center must request a non-delivery notification when sending the message.</p>
<b>Response elements</b>	(no parameters)

#### 9.9.5.2 send Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string1</language>
```





```

    <user
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string</user>
</soap:Header>
<soap:Body>
    <send xmlns="urn:IciMessageInterface">
        <messageId>string</messageId>
        <requestSuccessNotification>boolean</requestSuccessNotification>
        <requestFailureNotification>boolean</requestFailureNotification>
    </send>
</soap:Body>
</soap:Envelope>
    
```

### 9.9.5.3 sendResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <sendResponse xmlns="urn:IciMessageInterface">
            <response />
        </sendResponse>
    </soap:Body>
</soap:Envelope>
    
```

## 9.9.6 setActionFlags Operation

### 9.9.6.1 Definition

<b>Purpose</b>	The operation set the flags that indicate that the message has been read, answered and forwarded.
<b>Request name</b>	setActionFlags
<b>Request elements</b>	<b>String &lt;messageId&gt; : = itemId</b> <b>Boolean &lt;read&gt; : The message has been read.</b> <b>Boolean &lt;answered&gt; : The message has been answered.</b> <b>Boolean &lt;forwarded&gt; : The message has been forwarded.</b>
<b>Response elements</b>	(no parameters)

### 9.9.6.2 setActionFlags Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    
```



```
<soap:Header>
  <language
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string1</language>
  <user
xmlns="http://wicom.com/ws/OII/IciMessage/literalTypes">string</user>
</soap:Header>
<soap:Body>
  <setActionFlags xmlns="urn:IciMessageInterface">
    <messageId>string</messageId>
    <read>boolean</read>
    <answered>boolean</answered>
    <forwarded>boolean</forwarded>
  </setActionFlags>
</soap:Body>
</soap:Envelope>
```

### 9.9.6.3 setActionFlagsResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <setActionFlagsResponse xmlns="urn:IciMessageInterface">
      <response />
    </setActionFlagsResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.10 IciChatLine Interface

### 9.10.1 getSessions Operation

#### 9.10.1.1 Definition

<b>Purpose</b>	Gets the list of chat sessions assigned to a chat line. This method can be used for recovery purposes after the application has finished unexpectedly due to hardware failure, software failure or power outage.
<b>Request name</b>	getSessions
<b>Request elements</b>	<b>String &lt;chatLineId&gt; : = containerId.</b> Identifies a location where chat sessions are assigned to.
<b>Response elements</b>	<b>ChatSession[]&lt;chatSessions&gt;</b> : List of chat sessions assigned to the chat line.



### 9.10.1.2 getSession Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciChatLineInterface">ChatLineLanguage</language>
    <user xmlns="urn:IciChatLineInterface">ChatLineUser</user>
  </soap:Header>
  <soap:Body>
    <getSession xmlns="urn:IciChatLineInterface">
      <chatLineId>string</chatLineId>
    </getSession>
  </soap:Body>
</soap:Envelope>
```

### 9.10.1.3 getSessionResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getSessionResponse xmlns="urn:IciChatLineInterface">
      <response>
        <item>
          <title>string</title>
          <chatLineId>string</chatLineId>
          <chatSessionId>string</chatSessionId>
          <processingStatus>
            <id>string</id>
            <description>string</description>
          </processingStatus>
          <capabilityList>
            <item>int</item>
            <item>int</item>
          </capabilityList>
          <attachedData>string</attachedData>
          <chatStatus>
            <id>string</id>
            <description>string</description>
          </chatStatus>
          <chatParticipants>
            <item>string</item>
            <item>string</item>
          </chatParticipants>
        </item>
        <item>
          <title>string</title>
          <chatLineId>string</chatLineId>
          <chatSessionId>string</chatSessionId>
          <processingStatus>
            <id>string</id>
            <description>string</description>
          </processingStatus>
          <capabilityList>
            <item>int</item>
            <item>int</item>
          </capabilityList>
          <attachedData>string</attachedData>
```



```

        <chatStatus>
            <id>string</id>
            <description>string</description>
        </chatStatus>
        <chatParticipants>
            <item>string</item>
            <item>string</item>
        </chatParticipants>
    </item>
</response>
</getSessionsResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.11 IciChatSession Interface

### 9.11.1 getAttributes Operation

#### 9.11.1.1 Definition

<b>Purpose</b>	Gets all participants and other attributes of an existing chat session.
<b>Request name</b>	getAttributes
<b>Request elements</b>	<p><b>String &lt;chatSessionId&gt; : = itemId.</b> IM/Chat session ID.</p> <p><b>String &lt;chatLineId&gt; : = containerId.</b> Identifies a container for that chat session.</p>
<b>Response elements</b>	<b>ChatSession[]&lt;chatSession&gt;</b> : Data of that chat session.

#### 9.11.1.2 getAttributes Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="urn:IciChatSessionInterface">ChatSessionLanguage</language>
    <user xmlns="urn:IciChatSessionInterface">ChatSessionUser</user>
  </soap:Header>
  <soap:Body>
    <getAttributes xmlns="urn:IciChatSessionInterface">
      <chatSessionId>string</chatSessionId>
      <chatLineId>string</chatLineId>
    </getAttributes>
  </soap:Body>
    
```



```
</soap:Envelope>
```

### 9.11.1.3 getAttributesResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getAttributesResponse xmlns="urn:IciChatSessionInterface">
      <response>
        <title>string</title>
        <chatLineId>string</chatLineId>
        <chatSessionId>string</chatSessionId>
        <processingStatus>
          <id>string</id>
          <description>string</description>
        </processingStatus>
        <capabilityList>
          <item>int</item>
          <item>int</item>
        </capabilityList>
        <attachedData>string</attachedData>
        <chatStatus>
          <id>string</id>
          <description>string</description>
        </chatStatus>
        <chatParticipants>
          <item>string</item>
          <item>string</item>
        </chatParticipants>
      </response>
    </getAttributesResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.11.2 invite Operation

### 9.11.2.1 Definition

Purpose	<b>Initiates a chat session between any number of participants. An item of type chat session should be created via method Iciltem_create prior to this.</b>
Request name	<b>invite</b>
Request elements	<b>String &lt;chatSessionId&gt; : = itemId. IM/Chat session ID.</b> <b>String &lt;chatLineId&gt; : = containerId. Identifies a container for that chat session.</b> <b>String &lt;title&gt; : Title of chat session (optional).</b>



	<b>String[] &lt;chatParticipants&gt; : List of chat participants being invited (who did not accept the session yet).</b>
<b>Response elements</b>	<b>(no parameters)</b>

### 9.11.2.2 invite Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="urn:IciChatSessionInterface">ChatSessionLanguage</language>
    <user xmlns="urn:IciChatSessionInterface">ChatSessionUser</user>
  </soap:Header>
  <soap:Body>
    <invite xmlns="urn:IciChatSessionInterface">
      <chatSessionId>string</chatSessionId>
      <chatLineId>string</chatLineId>
      <title>string</title>
      <chatParticipants>
        <item>string</item>
        <item>string</item>
      </chatParticipants>
    </invite>
  </soap:Body>
</soap:Envelope>
```

### 9.11.2.3 inviteResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <inviteResponse xmlns="urn:IciChatSessionInterface">
      <response />
    </inviteResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.11.3 leave Operation

### 9.11.3.1 Definition

<b>Purpose</b>	Removes a chat line from chat session.
----------------	----------------------------------------



<b>Request name</b>	Leave
<b>Request elements</b>	<p><b>String &lt;chatSessionId&gt; : = itemId.</b> IM/Chat session ID.</p> <p><b>String &lt;chatLineId&gt; : = containerId.</b> Identifies a container for that chat session.</p>
<b>Response elements</b>	(no parameters)

### 9.11.3.2 leave Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="urn:IciChatSessionInterface">ChatSessionLanguage</language>
    <user xmlns="urn:IciChatSessionInterface">ChatSessionUser</user>
  </soap:Header>
  <soap:Body>
    <leave xmlns="urn:IciChatSessionInterface">
      <chatSessionId>string</chatSessionId>
      <chatLineId>string</chatLineId>
    </leave>
  </soap:Body>
</soap:Envelope>
```

### 9.11.3.3 leaveResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <leaveResponse xmlns="urn:IciChatSessionInterface">
      <response />
    </leaveResponse>
  </soap:Body>
</soap:Envelope>
```



## 9.11.4 getDialog Operation

### 9.11.4.1 Definition

<b>Purpose</b>	Gets all chat contributions created prior to method invocation.
<b>Request name</b>	getDialog
<b>Request elements</b>	<b>String &lt;chatSessionId&gt; : = itemId.</b> IM/Chat session ID.
<b>Response elements</b>	<b>ChatPosting[] &lt;chatPostings&gt;</b> : Represents the complete chat dialog till now.

### 9.11.4.2 getDialog Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="urn:IciChatSessionInterface">ChatSessionLanguage</language>
    <user xmlns="urn:IciChatSessionInterface">ChatSessionUser</user>
  </soap:Header>
  <soap:Body>
    <getDialog xmlns="urn:IciChatSessionInterface">
      <chatSessionId>string</chatSessionId>
    </getDialog>
  </soap:Body>
</soap:Envelope>
```

### 9.11.4.3 getDialogResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getDialogResponse xmlns="urn:IciChatSessionInterface">
      <response>
        <item>
          <chatSessionId>string</chatSessionId>
          <postDate>string</postDate>
          <chatParticipant>string</chatParticipant>
          <contentText>string</contentText>
          <systemMessage>boolean</systemMessage>
        </item>
        <item>
          <chatSessionId>string</chatSessionId>
          <postDate>string</postDate>
          <chatParticipant>string</chatParticipant>
        </item>
      </response>
    </getDialogResponse>
  </soap:Body>
</soap:Envelope>
```





```

        <contentText>string</contentText>
        <systemMessage>boolean</systemMessage>
    </item>
</response>
</getDialogResponse>
</soap:Body>
</soap:Envelope>
    
```

## 9.11.5 postMessage Operation

### 9.11.5.1 Definition

<b>Purpose</b>	Sends a chat contribution to the chat session.
<b>Request name</b>	postMessage
<b>Request elements</b>	<p><b>String &lt;chatSessionId&gt; : = itemId.</b> IM/Chat session ID.</p> <p><b>String &lt;chatLineId&gt; :</b> Identifies the chat line that makes the contribution.</p> <p><b>String &lt;contextText&gt; :</b> Post message as text.</p>
<b>Response elements</b>	(no parameters)

### 9.11.5.2 postMessage Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language
xmlns="urn:IciChatSessionInterface">ChatSessionLanguage</language>
    <user xmlns="urn:IciChatSessionInterface">ChatSessionUser</user>
  </soap:Header>
  <soap:Body>
    <postMessage xmlns="urn:IciChatSessionInterface">
      <chatSessionId>string</chatSessionId>
      <chatLineId>string</chatLineId>
      <contentText>string</contentText>
    </postMessage>
  </soap:Body>
</soap:Envelope>
    
```



### 9.11.5.3 postMessageResponse Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <postMessageResponse xmlns="urn:IciChatSessionInterface">
      <response />
    </postMessageResponse>
  </soap:Body>
</soap:Envelope>
```

## 9.12 IciActionItem Interface

### 9.12.1 route Operation

#### 9.12.1.1 Definition

<b>Purpose</b>	A given application object (for example, SAP actionItem) has to be routed and queued for one agent.
<b>Request name</b>	Route
<b>Request elements</b>	<p><b>String &lt;actionItemId&gt;</b> : Not used.</p> <p><b>ActionData &lt;itemData&gt;</b> : Contains actionItemAttributes and possibly requestId and processId (in CRM 5.2). This information should be returned in actionItemChanged event.</p> <p><b>Attribute[] &lt;routingAttributes&gt;</b> : Not used.</p> <p><b>String &lt;scenarioId&gt;</b> : Not used.</p>
<b>Response elements</b>	<b>String&lt;actionItemId&gt;</b> : Used in CRM 7.0 to identify the action item.

#### 9.12.1.2 route Message

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <language xmlns="urn:IciActionItemInterface">actionItemLanguage</language>
```



```

<user xmlns="urn:IciActionItemInterface">actionItemUser</user>
</soap:Header>
<soap:Body>
  <route xmlns="urn:IciActionItemInterface">
    <actionItemId>string</actionItemId>
    <itemData>
      <systemId>string</systemId>
      <client>string</client>
      <requestId>string</requestId>
      <processId>string</processId>
      <actionItemId>string</actionItemId>
      <actionItemInboxId>string</actionItemInboxId>
      <attachedData>string</attachedData>
      <capabilityList>
        <item>int</item>
        <item>int</item>
      </capabilityList>
      <processingStatus>
        <id>string</id>
        <description>string</description>
      </processingStatus>
      <actionItemStatus>
        <id>string</id>
        <description>string</description>
      </actionItemStatus>
      <actionItemAttributes>
        <item>
          <id>string</id>
          <description>string</description>
        </item>
        <item>
          <id>string</id>
          <description>string</description>
        </item>
      </actionItemAttributes>
    </itemData>
    <routingAttributes>
      <item>
        <id>string</id>
        <description>string</description>
      </item>
      <item>
        <id>string</id>
        <description>string</description>
      </item>
    </routingAttributes>
    <scenarioId>string</scenarioId>
  </route>
</soap:Body>
</soap:Envelope>

```

### 9.12.1.3 routeResponse Message

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <routeResponse xmlns="urn:IciActionItemInterface">
      <response>
        <actionItemID>string</actionItemID>
      </response>
    </routeResponse>

```



```
</soap:Body>
</soap:Envelope>
```

### 9.12.1.4 Sample Data

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <SOAP-ENV:Header>
    <pre:user xmlns:pre="urn:IciActionItemInterface" SOAP-ENV:mustUnderstand="0"
xsi:type="xsd:string">WF-BATCH</pre:user>
    <pre:language xmlns:pre="urn:IciActionItemInterface" SOAP-
ENV:mustUnderstand="0" xsi:type="xsd:string">EN</pre:language>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns0:route xmlns:ns0="urn:IciActionItemInterface">
      <itemData xmlns="urn:IciActionItemInterface">
        <systemId></systemId>
        <client></client>
        <requestId>000001210200</requestId>
        <processId>FOL2900000000004RAW3300000002161</processId>
        <attachedData></attachedData>
        <capabilityList></capabilityList>
        <processingStatus>
          <id></id>
          <description></description>
        </processingStatus>
        <actionItemStatus>
          <id></id>
          <description></description>
        </actionItemStatus>
        <actionItemAttributes>
          <item>
            <id>SUBJECT</id>
            <description>Action item test</description>
          </item>
          <item>
            <id>SUBJECT_CHARSET</id>
            <description></description>
          </item>
          <item>
            <id>SEND_DATE</id>
            <description>20080205111703 </description>
          </item>
          <item>
            <id>SENSITIVITY</id>
            <description></description>
          </item>
        </actionItemAttributes>
      </itemData>
    </ns0:route>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```

        <item>
            <id>IMPORTANCE</id>
            <description></description>
        </item>
        <item>
            <id>MAIL_ID</id>
            <description>2F6364050219C044A1303605B782EB5D05DE72F7@uspale20.pal.sap.corp</descripti
on>
        </item>
        <item>
            <id>FROM_NAME</id>
            <description>Thambala, Sudhir</description>
        </item>
        <item>
            <id>FROM_ADDRESS</id>
            <description>sudhir.thambala@sap.com</description>
        </item>
        <item>
            <id>TO_NAME</id>
            <description>ERMS@506.Q7D.R3.SAP-
AG.DE</description>
        </item>
        <item>
            <id>TO_ADDRESS</id>
            <description>ERMS@506.Q7D.R3.SAP-
AG.DE</description>
        </item>
    </actionItemAttributes>
</itemData>
<routingAttributes
xmlns="urn:IciActionItemInterface"></routingAttributes>
    <scenarioId xmlns="urn:IciActionItemInterface"></scenarioId>
</ns0:route>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```



# 10 ICI Content Description

This section describes the contents of the XML elements that carry task information from the Contact Center system via the interface.

## 10.1 TextElement Element

### 10.1.1 Definition

<b>Purpose</b>	The element is a global complex type that is used in other types. It contains a text ID and description.
<b>Element name</b>	TextElement
<b>Scope</b>	Global
<b>Sub-elements</b>	<b>String &lt;id&gt;</b> : The text ID. <b>String &lt;description&gt;</b> : The actual text.

### 10.1.2 XML Schema Description

```
<xs:complexType name="ici.TextElement">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="id" type="xs:string" nillable="true" minOccurs="0"/>
    <xs:element name="description" type="xs:string" nillable="true"
minOccurs="0"/>
  </xs:all>
</xs:complexType>
```

## 10.2 ProductInformation Element

### 10.2.1 Definition

<b>Purpose</b>	The element describes the OII version and the implementation to SAP.
----------------	----------------------------------------------------------------------



<b>Element name</b>	ProductInformation
<b>Scope</b>	IciSystem.exchangeProductInformation
<b>Sub-elements</b>	<p><b>String &lt;iciVersion&gt;</b> : The version of the Integrated Communication Interface that the external product was designed to support (“3.00”).</p> <p><b>String &lt;productName&gt;</b> : The name of the external product (“WicomOII”).</p> <p><b>String &lt;productVersion&gt;</b> : The version of the external product (no special structure of the version string required, for information only). For example, “1.0.2214.21309”.</p> <p><b>String &lt;iciMessageGroups&gt;</b> : The list of identifiers of those ICI groups that are supported by the external product. For example, “IciGroupBasic”, “IciGroupUser”, “IciGroupTelephony”, “IciGroupMessaging”.</p>

### 10.2.2 XML Schema Description

```

<xs:complexType name="ici.ProductInformation">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="productName" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="iciMessageGroups" type="s0:ArrayOfString"
nillable="true" minOccurs="0"/>
    <xs:element name="productVersion" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="iciVersion" type="xs:string" nillable="true"
minOccurs="0"/>
  </xs:all>
</xs:complexType>
    
```

## 10.3 QueueInfo Element

### 10.3.1 Definition

<b>Purpose</b>	To hold a Queue ID, its description and simple statistics.
<b>Element name</b>	QueueInfo



<b>Scope</b>	IciSystem.getPresenceQueueInfo
<b>Sub-elements</b>	<p><b>String &lt;id&gt;</b> : ID of Queue.</p> <p>Telephony : phone number of the queue                      Messaging: e-mail account name (?)                      Chat: chat address (e-mail address) of queue</p> <p><b>String &lt;description&gt;</b> : The display name of the queue.</p> <p><b>String &lt;loggedIn&gt;</b> : The number of logged in agents in queue.</p> <p><b>String &lt;ready&gt;</b> : The number of ready in agents in queue.</p> <p><b>String &lt;pending&gt;</b> : The number of items (phone calls) in queue.</p> <p><b>String &lt;open&gt;</b> : Is queue open = 1, closed = 0, empty = no info.</p> <p><b>String &lt;avgWaitingTime&gt;</b> : Average waiting time in seconds.</p> <p><b>String &lt;maxWaitingTime&gt;</b> : Maximum waiting time in seconds.</p> <p><b>String &lt;busy&gt;</b> : Number of busy agents.</p> <p><b>String &lt;wrapUp&gt;</b> : Number of wrap up agents.</p> <p><b>String &lt;notReady&gt;</b> : Number of not ready agents.</p> <p><b>String &lt;paused&gt;</b> : Number of paused agents.</p>

### 10.3.2 XML Schema Description

```

<s1:complexType name="QueueInfo">
  <s1:sequence>
    <s1:element minOccurs="0" maxOccurs="1" name="id" type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="description"
type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="loggedIn" type="s1:string"
/>
    <s1:element minOccurs="0" maxOccurs="1" name="ready" type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="pending" type="s1:string"
/>
    <s1:element minOccurs="0" maxOccurs="1" name="open" type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="avgWaitingTime"
type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="maxWaitingTime"
type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="busy" type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="wrapUp" type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="notReady"
type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="paused" type="s1:string"/>
  </s1:sequence>
</s1:complexType>
    
```





## 10.4 GroupInfo Element

### 10.4.1 Definition

<b>Purpose</b>	To hold a Group ID, its description, parent information and queues that they hold.
<b>Element name</b>	GroupInfo
<b>Scope</b>	IciSystem.getPresenceQueueInfo
<b>Sub-elements</b>	<p><b>String &lt;id&gt;</b> : ID of Group.</p> <p><b>String &lt;description&gt;</b> : Description of Group.</p> <p><b>String &lt;parentID&gt;</b> : ID of parent group.</p> <p><b>String &lt;queuesID&gt;</b> : List of presence queue IDs.</p>

### 10.4.2 XML Schema Description

```
<s1:complexType name="GroupInfo">
  <s1:sequence>
    <s1:element minOccurs="0" maxOccurs="1" name="id" type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="description"
type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="parentId" type="s1:string"
/>
    <s1:element minOccurs="0" maxOccurs="1" name="queuesId"
type="s0:ArrayOfString" />
  </s1:sequence>
</s1:complexType>
```

## 10.5 UserPresenceInfo Element

### 10.5.1 Definition

<b>Purpose</b>	To hold user ID, presence status and its description and collaboration information.
----------------	-------------------------------------------------------------------------------------



<b>Element name</b>	UserPresenceInfo
<b>Scope</b>	IciSystem.getUserPresence
<b>Sub-elements</b>	<p><b>String &lt;user&gt;</b> : User ID.</p> <p><b>String &lt;presenceStatus&gt;</b> :                  0 = NotAvailable                  1 = Available                  2 = Busy                  3 = LoggedOff</p> <p><b>String &lt;presenceDescription&gt;</b> : NotAvailable, Available, Busy, LoggedOff.</p> <p><b>CollaborationData[] &lt;collaborationData&gt;</b> : List of possible collaboration actions for the user.</p>

## 10.5.2 XML Schema Description

```
<s1:complexType name="ici.UserPresenceInfo">
  <s1:sequence>
    <s1:element minOccurs="1" maxOccurs="1" name="user" nillable="true"
type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="presenceStatus"
nillable="true" type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="presenceDescription"
nillable="true" type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="collaborationData"
nillable="true" type="s0:ArrayOfIciCollaborationData" />
  </s1:sequence>
</s1:complexType>
```

## 10.6 CollaborationData Element

### 10.6.1 Definition

<b>Purpose</b>	To hold the possible collaboration action for user.
<b>Element name</b>	CollaborationData
<b>Scope</b>	IciSystem.getUserPresence
<b>Sub-elements</b>	<p><b>String &lt;description&gt;</b> :                  Call</p>



	<p>Blind transfer Warm transfer</p> <p><b>String &lt;channel&gt; :</b> 1 = PhoneCall 2 = Message 3 = Chat 4 = Action</p> <p><b>String &lt;capability&gt; :</b> 101 = Dial 3 = Forward 104 = Transfer</p> <p><b>String &lt;address&gt; :</b> User phone number, e-mail address or chat address.</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 10.6.2 XML Schema Description

```
<s1:complexType name="ici.CollaborationData">
  <s1:sequence>
    <s1:element minOccurs="1" maxOccurs="1" name="description"
nillable="true" type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="channel" nillable="true"
type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="capability" nillable="true"
type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="address" nillable="true"
type="s1:string" />
  </s1:sequence>
</s1:complexType>
```

## 10.7 CallRecording Element

### 10.7.1 Definition

<b>Purpose</b>	Call recording information.
<b>Element name</b>	CallRecording
<b>Scope</b>	IciSystem.getCallRecordings
<b>Sub-elements</b>	<p><b>String &lt;callHandle&gt; :</b> Call handle.</p> <p><b>String &lt;userId&gt; :</b> User ID.</p>



	<p><b>DateTime &lt;creationTime&gt;</b> :When this recording was saved.</p> <p><b>DateTime &lt;startTime&gt;</b> :When the call was connected.</p> <p><b>DateTime &lt;endTime&gt;</b> :When the call was disconnected.</p> <p><b>String &lt;source&gt;</b> : Source phone number.</p> <p><b>String &lt;destination&gt;</b> : Destination phone number.</p> <p><b>String &lt;fileName&gt;</b> : File name of the call recording, containing the path if explicitly defined for this recording. Otherwise the path if the default recording path configured in SC. Explicit path is used if MRS has different recording path than the default.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 10.7.2 XML Schema Description

```

<s1:complexType name="CallRecording">
  <s1:sequence>
    <s1:element minOccurs="0" maxOccurs="1" name="callHandle"
type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="userId" type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="creationTime"
type="s1:dateTime" />
    <s1:element minOccurs="1" maxOccurs="1" name="startTime" nillable="true"
type="s1:dateTime" />
    <s1:element minOccurs="1" maxOccurs="1" name="endTime" nillable="true"
type="s1:dateTime" />
    <s1:element minOccurs="0" maxOccurs="1" name="source" type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="destination"
type="s1:string" />
    <s1:element minOccurs="0" maxOccurs="1" name="fileName" type="s1:string"
/ >
  </s1:sequence>
</s1:complexType>
    
```

## 10.8 CallBack Element

### 10.8.1 Definition

<b>Purpose</b>	Call back information.
<b>Element name</b>	CallBack
<b>Scope</b>	IciSystem.getCallbacks



<b>Sub-elements</b>	<p><b>String &lt;id&gt;</b> : Call back handle.</p> <p><b>String &lt;number&gt;</b> : Customer's phone number.</p> <p><b>String &lt;queueNumber&gt;</b> : Call back queue's number.</p> <p><b>DateTime &lt;handledTime&gt;</b> :When this call back was handled (result is SUCCESSFUL or HANDLED).</p> <p><b>DateTime &lt;lastCallTime&gt;</b> :When the call was last time done.</p> <p><b>DateTime &lt;nextCallTime&gt;</b> :When the call will be done next.</p> <p><b>Int &lt;failures&gt;</b> : Number of failures (result is BUSY, NO_ANSWER, BADADDRESS, VOICEMAIL, OTHER).</p> <p><b>Int &lt;maxCalls&gt;</b> : The maximum number of calls..</p> <p><b>String &lt;lastResult&gt;</b> : The last result (SUCCESSFUL, HANDLED or one of the failures).</p> <p><b>String &lt;extraData&gt;</b> : XML data for the call.</p> <p><b>String &lt;notes&gt;</b> : Free text for the call.</p> <p><b>String &lt;gwPrefix&gt;</b> : Gateway prefix that is used in routing.</p>
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 10.8.2 XML Schema Description

```

<s1:complexType name="CallBack">
  <s1:sequence>
    <s1:element minOccurs="0" maxOccurs="1" name="id" type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="number" type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="queueNumber"
type="s1:string"/>
    <s1:element minOccurs="1" maxOccurs="1" name="handledTime" nillable="true"
type="s1:dateTime"/>
    <s1:element minOccurs="1" maxOccurs="1" name="lastCallTime"
nillable="true" type="s1:dateTime"/>
    <s1:element minOccurs="1" maxOccurs="1" name="nextCallTime"
nillable="true" type="s1:dateTime"/>
    <s1:element minOccurs="1" maxOccurs="1" name="failures" type="s1:int"/>
    <s1:element minOccurs="1" maxOccurs="1" name="maxCalls" type="s1:int"/>
    <s1:element minOccurs="0" maxOccurs="1" name="lastResult"
type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="extraData"
type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="notes" type="s1:string"/>
    <s1:element minOccurs="0" maxOccurs="1" name="gwPrefix" type="s1:string"/>

```



```
</s1:sequence>
</s1:complexType>
```

## 10.9 Address Element

### 10.9.1 Definition

<b>Purpose</b>	The element holds a channel-specific address.
<b>Element name</b>	Address
<b>Scope</b>	User element
<b>Sub-elements</b>	<p><b>String &lt;channel&gt;</b> : The channel:</p> <p>1 = phone call</p> <p>2 = message</p> <p><b>String &lt;address&gt;</b> : The address. For example, “+123” or name@domain.com.</p>

### 10.9.2 XML Schema Description

```
<xs:complexType name="ici.Address">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="address" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="channel" type="xs:string" nillable="true"
minOccurs="0"/>
  </xs:all>
</xs:complexType>
```

## 10.10 User Element

### 10.10.1 Definition

<b>Purpose</b>	The element collects all attributes related to a user.
<b>Element name</b>	User



<b>Scope</b>	IciUser.getAttributes, IciEvent.userChanged
<b>Sub-elements</b>	<p><b>String &lt;userId&gt;</b> : Identifies the user. The user IDs must not contain slashes (the / character).</p> <p><b>TextElement &lt;currentWorkmode&gt;</b> : The current workmode of the user.</p> <p><b>TextElement[] &lt;workmodes&gt;</b> : The list of all workmodes the user is allowed to set:</p> <p>1 = Logged off</p> <p>2 = Logged on – ready</p> <p>3 = Logged on – not ready</p> <p>1000 = No Phone (treated as “Logged on – not ready”)</p> <p><b>Int &lt;wrapUpMode&gt;</b> : The wrap-up operation mode of the user:</p> <p>1 = automatic wrap up status</p> <p>2 = requested wrap up status</p> <p><b>TextElement[] &lt;currentChannels&gt;</b> : The list of channels enabled for this user.</p> <p><b>TextElement[] &lt;channels&gt;</b> : The list of channels the user is allowed to use:</p> <p>1 = phone call</p> <p>2 = message</p> <p><b>TextElement[] &lt;currentQueues&gt;</b> : The list of queues the user is currently assigned to.</p> <p><b>TextElement[] &lt;queues&gt;</b> : The list of queues the user is allowed to assign to.</p> <p><b>Address[] &lt;addresses&gt;</b> : The list of channel-specific addresses of the user.</p> <p><b>String &lt;cmdHostId&gt;</b> : Identifies the OII instance. Value is OII module GUID.</p>



## 10.10.2 XML Schema Description

```
<xs:complexType name="ici.User">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="userId" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="currentWorkmode" type="tns:ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="workmodes" type="s0:array_ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="wrapUpMode" type="xs:int"/>
    <xs:element name="currentChannels" type="s0:array_ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="channels" type="s0:array_ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="currentQueues" type="s0:array_ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="queues" type="s0:array_ici.TextElement" nillable="true"
minOccurs="0"/>
    <xs:element name="addresses" type="s0:array_ici.Address" nillable="true"
minOccurs="0"/>
    <xs:element minOccurs="1" maxOccurs="1" name="cmsHostId" nillable="true"
type="xs:string"/>
  </xs:all>
</xs:complexType>
```

## 10.11 Item Element

### 10.11.1 Definition

<b>Purpose</b>	The element collects all attributes that are common among all types of items.
<b>Element name</b>	Item
<b>Scope</b>	Icitem.create
<b>Sub-elements</b>	<p><b>String &lt;itemId&gt;</b> : The new GUID of the created item in the status “Created”.</p> <p><b>TextElement &lt;processingStatus&gt;</b> : The item processing status:</p> <p>1 = Not in process</p> <p><b>TextElement &lt;itemStatus&gt;</b> : The item status:</p>





	<p>1 = Created</p> <p><b>Int[] &lt;capabilityList&gt;</b> : The list of methods that are supported in the current “Created” status.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------

## 10.11.2 XML Schema Description

```

<xs:complexType name="ici.Item">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="status" type="tns:ici.TextElement" nillable="true"
minOccurs="0"/>
    <xs:element name="itemId" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="processingStatus" type="tns:ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="capabilityList" type="s0:ArrayOfint" nillable="true"
minOccurs="0"/>
  </xs:all>
</xs:complexType>
    
```

## 10.12 PhoneCall Element

### 10.12.1 Definition

<b>Purpose</b>	The element collects all attributes of a phone call.
<b>Element name</b>	PhoneCall
<b>Scope</b>	IciPhoneCall.getAttributes, IciEvent.phoneCallChanged
<b>Sub-elements</b>	<p><b>String &lt;callHandle&gt; : = itemId</b> : The GUID that identifies a call.</p> <p><b>String &lt;lineNumber&gt; : = containerId</b> : The number of the phone line the call is assigned to.</p> <p><b>TextElement &lt;processingStatus&gt;</b> : The item processing status:</p> <p>1 = Not in process</p> <p>2 = Active</p> <p>3 = Wrap up</p>



	<p>4 = Suspended</p> <p><b>TextElement &lt;callStatus&gt;</b> : The call status. See the list in the following section.</p> <p><b>Int[] &lt;capabilityList&gt;</b> : The capabilities valid for the current call status.</p> <p><b>String &lt;attachedData&gt;</b> : The data attached to the item.</p> <p><b>String[] &lt;remoteNumbers&gt;</b> : The table containing the phone numbers of all other parties (maybe more than one in case of conferences).</p> <p><b>String &lt;dialedNumber&gt;</b> : The dialed phone number.</p> <p><b>Boolean &lt;inbound&gt;</b> : Indicates the call direction.</p> <p><b>Boolean &lt;internal&gt;</b> : Indicates that the remote party is internal/external.</p> <p><b>String &lt;trunkId&gt;</b> : Identifies the trunk ID the call is routed through.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 10.12.2 Phonecall Capabilities

Id	Capability	Method
1	Accept (answer) a ringing call	IciItem_accept
2	Reroute a call before it is accepted	IciItem_reroute
3	Transfer a call to another destination number without consultation (is equal to deflect or blind transfer)	IciItem_forward
4	Indicate that wrap-up time is required	IciItem_wrapUpRequired
5	Indicate that wrap-up time is ended	IciItem_wrapUpEnded
6	Get attached data	IciItem_getAttachedData
7	Set attached data	IciItem_setAttachedData
8	Delete attached data	IciItem_deleteAttachedData
101	Dial (outbound call)	IciPhoneCall_dial
102	Disconnect a call	IciPhoneCall_drop
103	Create a consultation call	IciPhoneCall_consult



104	Transfer a held call to a consulted number	IciPhoneCall_transfer
105	Create a phone conference	IciPhoneCall_conference
106	Drop a party from a phone conference	IciPhoneCall_dropParty
107	Put a call on hold	IciPhoneCall_hold
108	Release a held call	IciPhoneCall_unhold
109	Toggle between a held and connected call	IciPhoneCall_alternate
110	Disconnect a connected call and release a held call	IciPhoneCall_reconnect
111	Listen to another call	IciPhoneCall_listen
112	End listening another call	IciPhoneCall_listenEnd

### 10.12.3 Phonecall Status

Id	Phone call status	Description	Default capabilities
1	Created	Initial outbound call created by an ICI application with Iciltem_create	6,7,8,101,102,103
2	New	Inbound call not assigned to any phone line (usually waiting in a queue)	
3	Alerting	Inbound call is signaled	1,2,3
4	Ended	Disconnected call	5
101	Connected	Call is connected, parties can talk	3,4,6,7,8,102,104,105,107,109,110
102	Idle	Initial call created by the contact center (e.g. user took the receiver off the hook)	6,7,8,101,102,103
103	Dialing	Outbound call is dialed	102
104	Ringing	Outbound call rings	102
105	Busy	Call is busy (the remote party is busy)	102
106	Hold	Call is held, on hold status was triggered by this number	4,6,7,8,102,103,104,105,108,109,110



107	Held	Call is held, on hold status was triggered by the remote number	4,6,7,8,102
108	ConfereceOwner	Call represents a conference call, established by this number	4,6,7,8,102,106,107
109	ConferencePartici pant	Call represents a conference call, established by a remote number	4,6,7,8,102,107
110	Listened	User (usually a supervisor) listens to a phone call	112

### 10.12.4 XML Schema Description

```

<xs:complexType name="ici.PhoneCall">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="lineNumber" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="callHandle" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="processingStatus" type="tns:ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="remoteNumbers" type="s0:ArrayOfString" nillable="true"
minOccurs="0"/>
    <xs:element name="dialedNumber" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="callStatus" type="tns:ici.TextElement" nillable="true"
minOccurs="0"/>
    <xs:element name="capabilityList" type="s0:ArrayOfint" nillable="true"
minOccurs="0"/>
    <xs:element name="inbound" type="xs:boolean"/>
    <xs:element name="internal" type="xs:boolean"/>
    <xs:element name="attachedData" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="trunkId" type="xs:string" nillable="true"
minOccurs="0"/>
  </xs:all>
</xs:complexType>
    
```

## 10.13 Message Element

### 10.13.1 Definition

<b>Purpose</b>	The element collects all attributes of an e-mail message.
<b>Element name</b>	Message



<b>Scope</b>	IciMessage.getAttributes, IciEvent.MessageChanged
<b>Sub-elements</b>	<p><b>String &lt;messageId&gt; : = itemId</b> : The GUID to identify a message.</p> <p><b>String &lt;folder&gt; : = containerId</b> : The folder where the message is located.</p> <p><b>TextElement &lt;processingStatus&gt;</b> : The item processing status:</p> <p>1 = Not in process</p> <p>2 = Active</p> <p>3 = Wrap up</p> <p>4 = Suspended</p> <p><b>TextElement &lt;messageStatus&gt;</b> : The message status. See the list in the following section.</p> <p><b>Int[] &lt;capabilityList&gt;</b> : The capabilities valid for the current message status.</p> <p><b>String &lt;attachedData&gt;</b> : The data attached to the item.</p> <p><b>String[] &lt;originator&gt;</b> : The originator’s address.</p> <p><b>String &lt;subject&gt;</b> : The subject.</p> <p><b>String &lt;messageText&gt;</b> : The contents of the main body text, not encoded.</p> <p><b>String[] &lt;attachments&gt;</b> : The attachment names.</p> <p><b>Boolean &lt;notification&gt;</b> : Indicates that the message is a notification.</p> <p><b>String &lt;originalMessageId&gt;</b> : The item ID of message the notification is referring to (filled only if the message is a notification).</p> <p><b>Boolean &lt;read&gt;</b> : Indicates that the message has been read.</p> <p><b>Boolean &lt;answered&gt;</b> : Indicates that the message has been answered.</p> <p><b>Boolean &lt;forwarded&gt;</b> : Indicates that the message has been forwarded.</p>



### 10.13.2 Message Capabilities

Id	Capability	Method
1	Accept a new message for processing	Icitem_accept
2	Reroute a message before it is accepted	Icitem_reroute
3	Forward a message to another destination	Icitem_forward
4	Indicate that wrap-up time is required	Icitem_wrapUpRequired
5	Indicate that wrap-up time is ended	Icitem_wrapUpEnded
6	Get attached data	Icitem_getAttachedData
7	Set attached data	Icitem_setAttachedData
8	Delete attached data	Icitem_deleteAttachedData
201	Set message contents	IciMessage_setContent
202	Send message	IciMessage_send
203	Delete message	IciMessage_delete

### 10.13.3 Message Status

Id	Message status	Description	Default capabilities
1	<i>Created</i>	<i>Initial message created by an ICI application with Icitem_create</i>	6,7,8,201
2	New	Inbound message not assigned to an agent (usually waiting in a queue)	
3	Alerting	Inbound message signaled to an agent	1,2,3
4	Ended	Message deleted (may be archived)	5
201	Accepted	Inbound message accepted by an agent	3, 4, 6, 7, 8, 203



202	Draft	Message has been saved but not yet sent	4, 6, 7, 8, 201, 202, 203
203	Sent	Message has been sent	4, 5, 6, 7, 8, 203

### 10.13.4 XML Schema Description

```
<xs:complexType name="ici.Message">
  <xs:all>
    <xs:element name="message" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="folder" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="subject" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="messageId" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="processingStatus" type="tns:ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="originator" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="messageText" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="attachments" type="s0:ArrayOfString" nillable="true"
minOccurs="0"/>
    <xs:element name="messageStatus" type="tns:ici.TextElement"
nillable="true" minOccurs="0"/>
    <xs:element name="capabilityList" type="s0:ArrayOfint" nillable="true"
minOccurs="0"/>
    <xs:element name="attachedData" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="originalMessageId" type="xs:string" nillable="true"
minOccurs="0"/>
    <xs:element name="notification" type="xs:boolean"/>
    <xs:element name="read" type="xs:boolean"/>
    <xs:element name="answered" type="xs:boolean"/>
    <xs:element name="forwarded" type="xs:boolean"/>
  </xs:all>
</xs:complexType>
```

## 10.14 ChatSession Element

### 10.14.1 Definition

<b>Purpose</b>	The element collects all attributes of an e-mail message.
<b>Element name</b>	Message
<b>Scope</b>	IciChatSession.getAttributes, IciEvent.chatChanged



<b>Sub-elements</b>	<p><b>String &lt;chatSessionId&gt; := itemId</b> : The GUID to identify a chat session.</p> <p><b>String &lt;chatLineId&gt; := containerId</b> : ID of the chat line to which this session is assigned to. This is the chat address of the agent, in e-mail address format.</p> <p><b>TextElement &lt;processingStatus&gt;</b> : The item processing status:</p> <p>1 = Not in process</p> <p>2 = Active</p> <p>3 = Wrap up</p> <p>4 = Suspended</p> <p><b>TextElement &lt;chatStatus&gt;</b> : Chat status. See the list in the following section.</p> <p><b>Int[] &lt;capabilityList&gt;</b> : The capabilities valid for the current chat status.</p> <p><b>String &lt;attachedData&gt;</b> : Item attached data.</p> <p><b>String &lt;title&gt;</b> : Title of this chat session (may be empty).</p> <p><b>String[] &lt;chatParticipants&gt;</b> : List of chat participants. It does not contain assigned users who did not accept till now.</p>
---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 10.14.2 ChatSession Capabilities

Id	Capability	Method
1	Accept a new chat session for processing	Icitem_accept
2	Reroute a chat session before it is accepted	Icitem_reroute
3	Forward a chat session to another destination	Icitem_forward
4	Indicate that wrap-up time is required	Icitem_wrapUpRequired
5	Indicate that wrap-up time is ended	Icitem_wrapUpEnded
6	Get attached data	Icitem_getAttachedData





7	Set attached data	IciItem_setAttachedData
8	Delete attached data	IciItem_deleteAttachedData
301	Leave a chat session	IciChatSession_leave
302	Join an existing chat session not signaled via event (not implemented)	IciChatSession_join
303	Start a conference chat session (not implemented)	IciChatSession_conference
304	Post a message to the chat session	IciChatSession_postMessage
305	Invite participants in a chat session	IciChatSession_invite

### 10.14.3 ChatSession Status

<b>Id</b>	<b>Chat Session status</b>	<b>Description</b>	<b>Default capabilities</b>
1	Created	Initial chat session created by an ICI application with IciItem_create	6,7,8,302,305
2	New	Chat session not assigned to any chat line (usually waiting in a queue)	
3	Alerting	Inbound chat session is signaled	1,2,3
4	Ended	Ended chat session	5
301	Connected	Chat session is connected, parties can talk	3,4,6,7,8,301,302,303,304
302	Inviting	Chat session participants are being invited (equivalent to status “ringing” in telephony)	301

### 10.14.4 XML Schema Description

```
<s1:complexType name="ici.ChatSession">
  <s1:sequence>
```



```

<s1:element minOccurs="1" maxOccurs="1" name="title" nillable="true"
type="s1:string" />
<s1:element minOccurs="1" maxOccurs="1" name="chatLineId" nillable="true"
type="s1:string" />
<s1:element minOccurs="1" maxOccurs="1" name="chatSessionId" nillable="true"
type="s1:string" />
<s1:element minOccurs="1" maxOccurs="1" name="processingStatus" nillable="true"
type="s0:ici.TextElement" />
<s1:element minOccurs="1" maxOccurs="1" name="capabilityList" nillable="true"
type="s0:ArrayOfInt" />
<s1:element minOccurs="1" maxOccurs="1" name="attachedData" nillable="true"
type="s1:string" />
<s1:element minOccurs="1" maxOccurs="1" name="chatStatus" nillable="true"
type="s0:ici.TextElement" />
<s1:element minOccurs="1" maxOccurs="1" name="chatParticipants" nillable="true"
type="s0:ArrayOfString" />
</s1:sequence>
</s1:complexType>
    
```

## 10.15 ChatPosting Element

### 10.15.1 Definition

<b>Purpose</b>	Attributes of one posting to a chat session.
<b>Element name</b>	ChatPosting
<b>Scope</b>	IciChatSession.postMessage, IciEvent.chatNewPosting
<b>Sub-elements</b>	<p><b>String &lt;chatSessionId&gt; : = itemId</b> : ID of chat session to which this posting was sent.</p> <p><b>String &lt;postDate&gt;</b> : Time stamp of posting (YYYYMMDDHHMMSS).</p> <p><b>String &lt;chatParticipant&gt;</b> : Chat participant, originator of the posting. This is the chat address of agent or customer, in e-mail address format.</p> <p><b>String &lt;contentText&gt;</b> : Message that was posted.</p> <p><b>Boolean &lt;systemMessage&gt;</b> : Indicates a posting generated by the chat server (in this case chatParticipant may be empty).</p>

### 10.15.2 XML Schema Description

```

<s1:complexType name="ici.ChatPosting">
  <s1:sequence>
    <s1:element minOccurs="1" maxOccurs="1" name="chatSessionId" nillable="true"
    
```



```

type="s1:string" />
  <s1:element minOccurs="1" maxOccurs="1" name="postDate" nillable="true"
type="s1:string" />
  <s1:element minOccurs="1" maxOccurs="1" name="chatParticipant" nillable="true"
type="s1:string" />
  <s1:element minOccurs="1" maxOccurs="1" name="contentText" nillable="true"
type="s1:string" />
  <s1:element minOccurs="1" maxOccurs="1" name="systemMessage" type="s1:boolean" />
</s1:sequence>
</s1:complexType>
    
```

## 10.16 ActionSession Element

### 10.16.1 Definition

<b>Purpose</b>	Collects status of an action item. See also ActionItem element.
<b>Element name</b>	ActionSession
<b>Scope</b>	IciEvent.actionItemChanged (IciActionItem.getAttributes)
<b>Sub-elements</b>	<p><b>String[] &lt;actionParticipants&gt;</b> : List of user ids. Contains the id of the active user, if any.</p> <p><b>Int[] &lt;capabilityList&gt;</b> : The capabilities valid for the current action item status.</p> <p><b>String &lt;title&gt;</b> : Title of this action item session. Currently not used.</p> <p><b>String &lt;actionSessionId&gt;</b> : The GUID to identify a chat session. Currently not used.</p> <p><b>TextElement &lt;actionStatus&gt;</b> : Action item status. See the list in the following section.</p> <p><b>TextElement &lt;processingStatus&gt;</b> : The item processing status:</p> <ul style="list-style-type: none"> <li>1 = Not in process</li> <li>2 = Active</li> <li>3 = Wrap up</li> <li>4 = Suspended</li> </ul>



	<p><b>String &lt;attachedData&gt;</b> : Item attached data.</p> <p><b>String &lt;actionLineId&gt;</b> : ID of the action item line to which this session is assigned to. Currently not used.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 10.16.2 ActionSession Capabilities

Id	Capability	Method
1	Accept a new action item for processing	IciItem_accept
2	Reroute an action item before it is accepted	IciItem_reroute
3	Forward an action item to another destination	IciItem_forward
4	Indicate that wrap-up time is required	IciItem_wrapUpRequired
5	Indicate that wrap-up time is ended	IciItem_wrapUpEnded
6	Get attached data	IciItem_getAttachedData
7	Set attached data	IciItem_setAttachedData
8	Delete attached data	IciItem_deleteAttachedData
401	Enqueue (not implemented)	IciActionItem_enqueue
402	Route (this actually creates the action item, so this capability is not used)	IciActionItem_route
403	End	IciMessage_end (was IciActionItem_end)

### 10.16.3 ActionSession Status

Id	Action Session status	Description	Default capabilities



1	Created	Action Item created by an ICI application with Icilitem_create (not used, instead IciActionItem_route is used)	6,7,8,401,402,403
2	New	Action Item not assigned to an agent (usually waiting in a queue)	
3	Alerting	Action Item signaled to an agent	1,2,3,6
4	Ended	Ended Action Item	5
401	Accepted	Action Item accepted by an agent	3, 4, 6, 7, 8, 403

### 10.16.4 XML Schema Description

```
<s1:complexType name="ici.ActionSession">
  <s1:sequence>
    <s1:element minOccurs="1" maxOccurs="1" name="actionParticipants" nillable="true"
type="s0:ArrayOfString1" />
    <s1:element minOccurs="1" maxOccurs="1" name="capabilityList" nillable="true"
type="s0:ArrayOfInt" />
    <s1:element minOccurs="1" maxOccurs="1" name="title" nillable="true"
type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="actionSessionId" nillable="true"
type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="actionStatus" nillable="true"
type="s0:ici.TextElement" />
    <s1:element minOccurs="1" maxOccurs="1" name="processingStatus" nillable="true"
type="s0:ici.TextElement" />
    <s1:element minOccurs="1" maxOccurs="1" name="attachedData" nillable="true"
type="s1:string" />
    <s1:element minOccurs="1" maxOccurs="1" name="actionLineId" nillable="true"
type="s1:string" />
  </s1:sequence>
</s1:complexType>
```

## 10.17 ActionItem Element

### 10.17.1 Definition

<b>Purpose</b>	Collects all attributes of an action item.  <b>NOTE:</b> Some of the attributes are identical to attributes in ActionSession element.
<b>Element name</b>	ActionItem



<b>Scope</b>	IciActionItem_route, IciEvent.actionItemChanged (IciActionItem.getAttributes)
<b>Sub-elements</b>	<p><b>String &lt;systemId&gt;</b> : Not used.</p> <p><b>String &lt;client&gt;</b> : Not used.</p> <p><b>String &lt;requestId&gt;</b> : Used in CRM 5.2 to identify the action item. Not used in CRM 7.0.</p> <p><b>String &lt;processId&gt;</b> : Used in CRM 5.2 to identify the action item. Not used in CRM 7.0.</p> <p><b>String &lt;actionItemId&gt;</b> : The GUID to identify the action item. Currently not used.</p> <p><b>String &lt;actionItemInboxId&gt;</b> : ID of the action item line to which this session is assigned to. Currently not used.</p> <p><b>String &lt;attachedData&gt;</b> : Item attached data. See ActionSession.</p> <p><b>Int[] &lt;capabilityList&gt;</b> : The capabilities valid for the current action item status. See ActionSession.</p> <p><b>TextElement &lt;processingStatus&gt;</b> : The item processing status. See ActionSession.</p> <p><b>TextElement &lt;actionItemStatus&gt;</b> : Action item status. See ActionSession.</p> <p><b>Attribute &lt;actionItemAttributes&gt;</b> : Attributes from route operation, internal to CRM.</p>

## 10.17.2 XML Schema Description

```

<sl:complexType name="ici.ActionItem">
  <sl:sequence>
    <sl:element minOccurs="1" maxOccurs="1" name="systemId" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="client" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="requestId" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="processId" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="actionItemId" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="actionItemInboxId" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="attachedData" nillable="true"
type="s1:string" />
    <sl:element minOccurs="1" maxOccurs="1" name="capabilityList" nillable="true"
type="s0:ArrayOfInt" />
  
```



```

<s1:element minOccurs="1" maxOccurs="1" name="processingStatus" nillable="true"
type="s0:ici.TextElement" />
<s1:element minOccurs="1" maxOccurs="1" name="actionItemStatus" nillable="true"
type="s0:ici.TextElement" />
<s1:element minOccurs="1" maxOccurs="1" name="actionItemAttributes"
nillable="true" type="s0:ArrayOfIciAttribute" />
</s1:sequence>
</s1:complexType>
    
```

## 10.18 Attribute Element

### 10.18.1 Definition

<b>Purpose</b>	General Attribute element used in ActionItem element.
<b>Element name</b>	Attribute
<b>Scope</b>	IciActionItem_route, IciEvent.actionItemChanged (IciActionItem.getAttributes)
<b>Sub-elements</b>	<b>String &lt;id&gt;</b> : Name of attribute. <b>String &lt;description&gt;</b> : Value of attribute.

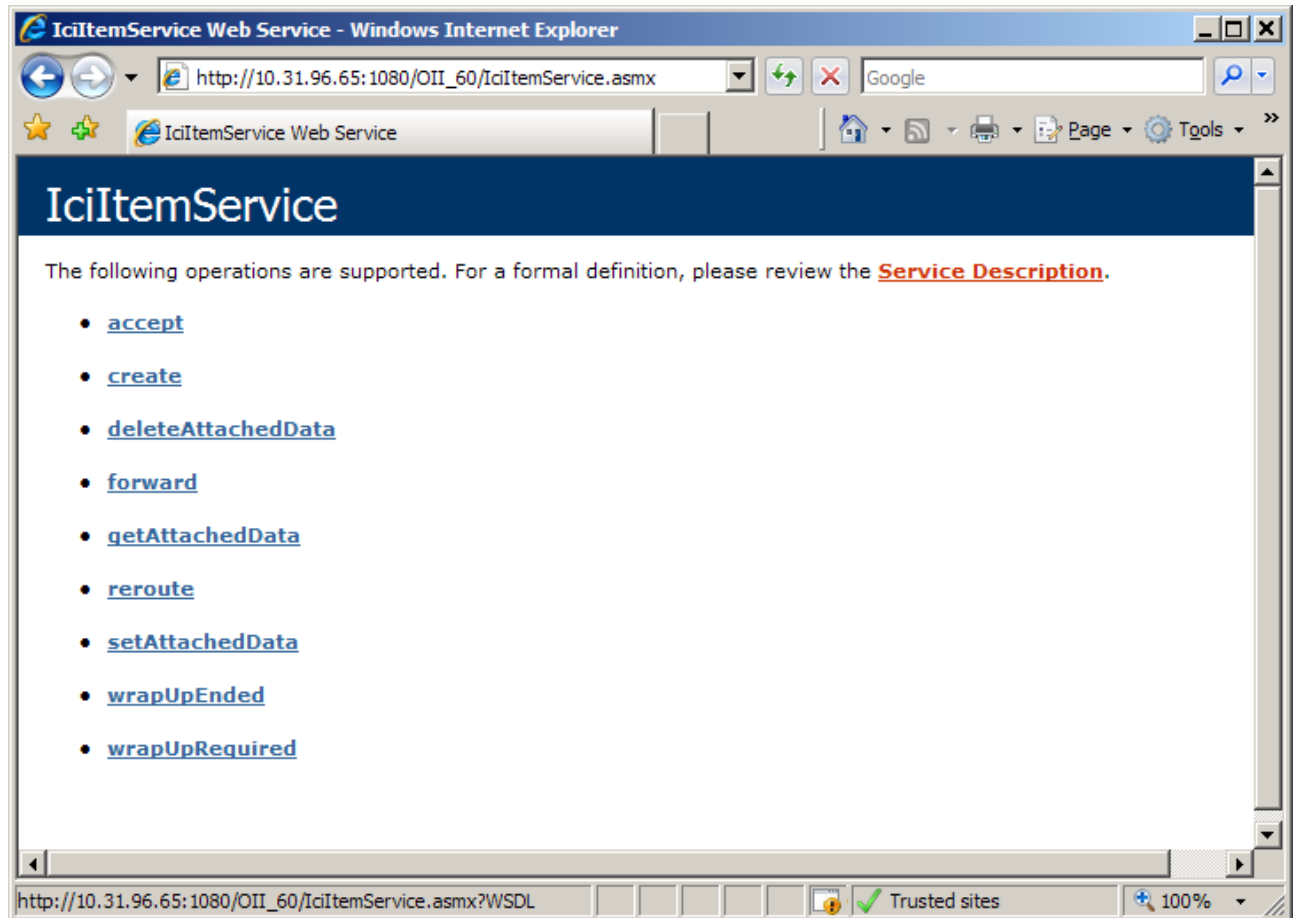
### 10.18.2 XML Schema Description

```

<s1:complexType name="ici.Attribute">
  <s1:sequence>
    <s1:element minOccurs="1" maxOccurs="1" name="id" nillable="true" type="s1:string"
    />
    <s1:element minOccurs="1" maxOccurs="1" name="description" nillable="true"
    type="s1:string" />
  </s1:sequence>
</s1:complexType>
    
```

# 11 ICI WSDL Description

The WSDL (<http://www.w3.org/TR/wsd/>) description of the interface is available on request. WSDL is also available automatically in each separate web service, like Iciltem, by giving the ?WSDL parameter to the web service.



For example <http://ip-address/OII/IciltemService.asmx?WSDL> shows the Iciltem WSDL in browser.







## 12 Troubleshooting

The OII log file is in the same directory as other log files (for example, C:\Program files\SAP\BCM\VU\VU1\logs) and its name has the following syntax:

OII\_VU1\_yyyymmdd.log.

The beginning of file contains version and configuration information:

```

-----
2011-08-22 (Monday, 22 August 2011)
SAP Contact Center OII 7.0.1.10033 1.0.4203.763 2011-07-05 01:25:26Z
file:///D:/Program Files/SAP/BCM/VU/VU_OII/web/OII/bin/WicomOII.DLL
Domain /LM/W3SVC/3/ROOT/OII-3-129584935990382311 with ID = 4.
Process id=3472 name=w3wp working set=101MB private=85MB handles=1044 tot
cpu=00:03:03.2343750 user cpu=00:02:38.4375000
VU=VU_OII UseSecureLogging=False TickCount=212041812
_MAP_SUBSCRIBE_URL =
IpPort = 21097
LOG_DETECT_REPEATED = 1
LOG_KEEP_DAYS = 7
LOG_LEVEL = 3
LOG_SHOW_THREAD = 1
MAP_SUBSCRIBE_URL =
----ENV START---- 40
ALLUSERSPROFILE=C:\ProgramData
...
----ENV END----
Loaded = 2011-08-22 13:33:20.971
Started = 0001-01-01 00:00:00.000
WinUser = DEWDFIAS406\comadmin

Delete old logfile: D:\Program
Files\SAP\BCM\VU\VU_OII\logs\OII_VU_OII_20110815.log
13:33:21.013 INF> [Translator] 15 languages 38 rows in D:\Program
Files\SAP\BCM\VU\VU_OII\web\OII\translator.txt
13:33:21.014 INF> Preview
13:33:21.015 INF> english : Preview
...
13:33:21.023 INF> AppState: Uninitialized => Stopping
13:33:21.023 INF> Stop... reason=Start
LOG STATS: EXC:0 ERR:0 INF:19 TRC:0 DBG:0
13:33:21.034 INF> AppState: Stopping => Stopped
13:33:21.034 INF> Stop complete, reason=Start
13:33:21.052 INF> [CEM] Server=DEWDFIAS406 Database=DEWDFIAS406
User=DEWDFIAS406\comadmin Date=2011-08-22 15:33:21.050 UTC=2011-08-22
13:33:21.050
13:33:21.055 INF> new LibIPCPeer
13:33:21.068 INF> AddLocalPeer Oii:VU_OII=10.66.193.43:21097
13:33:21.109 INF> OII parameters for VU=VU_OII
13:33:21.110 INF> guid=29168c15-a63c-49b0-a646-bd5131048b47 ip=10.66.193.43
13:33:21.110 INF> IpPort=21097
13:33:21.110 INF> OII.Action=1
...

```



Each line after that contains the time, the debugging level, the thread name, and the actual log information. The EXC> and ERR> debugging levels are exceptions and errors which should not happen during normal operation.

```
13:35:22.648 TRC> (Status) State=Connected Chat=Connected Clients=1 Users=1
Items=0 Subs=0 [SOAP:0/0/0/25 q=0/0 ms=0/0/0 #=0] [Chat:0/0/0/25 q=0/0
ms=0/0/0] SAPphone:1=Running;2=Running;3=Running;
Oii:VU_OII=False;ChatServer:VU_OII=True;AgentServer:VU_ADMIN=True;AgentServ
er:VU_SOFTPHONES=True;
```

The example line above contains the OII status and statistical information about the connection with SAP CRM, and it is written once a minute.

- State=Connected : Indicates that the interface is connected to AS.
- Chat=Connected : Indicates that the interface is connected to Chat Server.
- Clients=1 : Indicates that there are currently 1 phone clients (CDTs) connected to AS.
- Users=1 : Indicates that there are 1 users cached in OII memory.
- Items=0 : Indicates that currently there are 0 items (phone calls, chats or e-mail messages) in the OII memory.
- Subs=0 : Indicates that there are 0 active subscriptions from SAP CRM.
- [SOAP:0/0/0/25 q=0/0 ms=0/0/0 #=0] : Statistics for SOAP events sent to CRM. Indicates the following:
  - SOAP:0/0/0/25
    - No notifications in progress (0).
    - The maximum number of simultaneous notifications (0).
    - Number of threads (0).
    - Maximum number of threads (25).
  - q=0/0
    - No items in the notification queue (0).
    - The maximum number of notifications in the queue (0)
  - ms=0/0/0



- The minimum time for a notification (SOAP call from Chat Portal to ICC) is 0 ms.
- The average time is 0 ms
- The maximum time is 0 ms.

1. #=0

- The number of notifications so far is 0.
- There are no errors.
- [Chat:0/0/0/25 q=0/0 ms=0/0/0] : Same as [SOAP] but this is for Chat Server connection.



## 13 Glossary

<b>Term</b>	<b>Description</b>
CEM	Contact Center service that handles phone calls and e-mail tasks
CDT	Contact Center Communication Desktop application running on individual client workstations
ChP	Contact Center Chat Portal
ChS	Contact Center Chat Server
CoS	Contact Center Connection Server
CRM	Customer Relationship Management
IA	Contact Center Infrastructure Administrator
IC	SAP Interaction Center
ICI	SAP Integrated Communication Interface
IIS	Microsoft Internet Information Services
OII	Contact Center Online Interaction Interface
SC	Contact Center System Configurator
SOAP	Simple Object Access Protocol
WACP	Contact Center network protocol for TCP/IP messages in simple key=url_encoded_value; format.

