

# VersaDev

134 Gilbert Street Adelaide South Australia 5000

t:: +61.8.8463 1914

f:: +61.8.8212 8447

e:: info@versadev.com

ABN: 53 087 184 807

**versaSRS**

## **Phone System Integration Guide**

This is an unpublished work the copyright of which vests in VersaDev Pty Ltd. All rights reserved. The information contained herein is the property of VersaDev Pty Ltd and is supplied without liability for errors or omissions and no part may be reproduced or disclosed except as authorised by prior written permission. The copyright and foregoing restriction on reproduction use and disclosure extend to all the media in which the information may be embodied, including magnetic or electronic storage.

## Document Control Information

It is the reader's responsibility to ensure that they have the latest version of this document. Questions should be directed to the author(s) of this document.

Revision	Date	Revised By	Description
1.0	28/04/2009		Initial Issue
2.0	19/05/2009	Warren Bullock	Additional information

**Document Sign Off**

Prepared, Review and Approved by			Signature	Date (dd/mm/yyyy)
Prepared By	Chief Software Architect	Warren Bullock		
Reviewed By				
Approved By				

## Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>5</b>
<b>2</b>	<b>OVERVIEW.....</b>	<b>5</b>
<b>3</b>	<b>VERSASRS ENHANCEMENTS TO SUPPORT PHONE SYSTEM INTEGRATION.....</b>	<b>5</b>
3.1	REQUEST TO OPEN AN INVALID CALL .....	9
3.2	REQUEST TO OPEN A CALL LOCKED BY ANOTHER OPERATOR.....	9
3.3	REQUEST TO OPEN AN EXISTING CALL WHERE THE OPERATOR IS NOT A MEMBER OF THE QUEUE.....	10
3.4	PHONE ROUTE LOGGING .....	10
<b>4</b>	<b>SYSTEM REQUIREMENTS .....</b>	<b>10</b>

# 1 Introduction

This document outlines the enhancements that have been made to versaSRS so that the system can integrate with a soft phone capable of launching a URL when an incoming call is received.

## 2 Overview

versaSRS is a thin-client Web browser delivered service request / ticket management system. Tickets are created manually by system operators or automatically by emailing directly in to the system. Each ticket contains the contact details of the requestor, typically first name, surname, telephone number and email address. Contact details are obtained from the contacts module or manually entered by a system operator. The contacts module is populated by manually entering data or by importing data from other contact repositories such as Active Directory or Customer Relationship Management (CRM) systems.

System operators that manually enter new tickets generally receive a new request via a telephone call. Requestors may follow up these requests with a telephone call. Therefore, to improve this process versaSRS has been enhanced to provide an interface to allow phone systems to integrate with the system. There are now many different phone systems to choose from. Many of these phone systems offer a soft phone or desktop software for displaying incoming caller identification or the ability to park and transfer calls. Typically these phone systems could be integrated in to versaSRS by developing desktop based solutions that consume software libraries available from the provider of the phone system. However, VersaDev would need to update these solutions as vendors make changes to their phone systems and APIs. This would be very time consuming and require a lot of VersaDev's resources to develop and support this. Therefore, we have enhanced versaSRS to allow either a soft phone or desktop software to integrate with the system by launching a URL. Provided the phone system software can launch a URL or be easily extended to do so, the calling software will be able to expose ticket management functionality available within versaSRS. It will be the responsibility of the vendor of the phone system to provide this functionality. The enhancements to versaSRS to support this functionality are presented below.

## 3 versaSRS Enhancements to Support Phone System Integration

versaSRS has been enhanced to support the integration of phone systems capable of launching a URL. An example URL is provided below and shows the available parameters:

<http://myVersaSRS/phonetools.aspx?userentrymode=3&userentry=31481&keyPress=551256&callerID=9876>

Parameter	Description
userentrymode	Integer value defining how the ticket contact details are to be determined. Value = 1 - perform contact lookup based on supplied email address Value = 2 - perform contact lookup based on supplied username Value = 3 - perform contact lookup based on supplied Employee Id
userentry	Defines the data to be used to perform either a contact lookup or the unique versaSRS Ticket number to be used to display an existing record to the operator.  <b>Note:</b> when performing a contact lookup, the value passed by userentry is either the requestor's email address, username or employee id. The method of lookup will be governed by the value of userentrymode.
keyPress	String value – unique code passed by the phone system. This is the Vector Directory Number (VDN) for an Avaya Phone System. The value of keyPress determines the behavior of the system. See PhoneRoutes.xml.
callerID	String value – currently not used, reserved for future development.

The phone system software must be configured to launch the required URL and pass the appropriate values for each parameter.

versaSRS has been extended to include an XML file (PhoneRoutes.xml). This file determines how the information passed by the URL is processed by the system. A sample file is presented below:

```
<?xml version="1.0" encoding="utf-8" ?>
<PhoneRoutes>
  <RouteCRMBaseURL>Modules/CRM_Contacts/CRM_Search.aspx?mode=1</RouteCRMBaseURL>
  <RouteKeyPressRejectLength>7</RouteKeyPressRejectLength>
  <RouteContactQueries>
    <RouteContactQuery1></RouteContactQuery1>
    <RouteContactQuery2></RouteContactQuery2>
    <RouteContactQuery3>SELECT CID AS ContactID, Company AS CField1, FirstName
AS CField2, Surname AS CField3, BEmail AS CField4, BPhone AS CField5, BDivision
AS CField6, BDepartment AS CField7 FROM tbl_Contacts WHERE CID =
[CONTACTLOOKUPKEY]</RouteContactQuery3>
  </RouteContactQueries>
  <Routes>
    <item>
      <key>34</key>
      <template>10</template>
      <userentrytype>2</userentrytype>
      <showcontacts>1</showcontacts>
      <showkbkeysets>0</showkbkeysets>
      <nocall>0</nocall>
    </item>
  </item>
</Routes>
</PhoneRoutes>
```

```

        <key>56</key>
        <template>15</template>
        <userentrytype>2</userentrytype>
        <showcontacts>1</showcontacts>
        <showkbkeysets>1</showkbkeysets>
        <nocall>0</nocall>
    </item>
    <item>
        <key>84</key>
        <template>13</template>
        <userentrytype>2</userentrytype>
        <showcontacts>1</showcontacts>
        <showkbkeysets>0</showkbkeysets>
        <nocall>0</nocall>
    </item>
    <item>
        <key>09</key>
        <template>0</template>
        <userentrytype>2</userentrytype>
        <showcontacts>1</showcontacts>
        <showkbkeysets>0</showkbkeysets>
        <nocall>1</nocall>
    </item>
</Routes>
</PhoneRoutes>
    
```

Each of the key nodes in the XML document is described below:

Node	Description
RouteCRMBaseURL	Defines the URL of the CRM / Contacts module. This allows versaSRS to launch the appropriate CRM / Contacts module when a new telephone call is answered.
RouteKeyPressRejectLength	Defines the maximum number of characters - 1 that can be received for the parameter keyPress before the system will reject the request. When this occurs the Web browser window that was launched is closed. When this occurs the phone route request is not logged.
RouteContactQuery1	Defines the SQL Contact Lookup Query to be used when userentrymode = 1
RouteContactQuery2	Defines the SQL Contact Lookup Query to be used when userentrymode = 2
RouteContactQuery3	Defines the SQL Contact Lookup Query to be used when userentrymode = 3
Item	Defines the start of a phone route
key	A unique value that corresponds to the keyPress passed by the phone system
template	The versaSRS Call template id. This template will be applied if the phone route defines the creation of a new Call, governed by userentrytype.
userentrytype	userentrytype = 1 - display an existing Call. userentrytype = 2 - create a new Call based on a Call template.  <b>Note:</b> when userentrytype = 1, the application assumes that the value of

	<p>userentry is an existing versaSRS Ticket number.</p>
showcontacts	<p>showcontacts = 0 – do not display the CRM / Contacts module.                  showcontacts = 1 – display the CRM / Contacts module.</p> <p><b>Note:</b> the CRM / Contacts module will default to display the contact details defined by the value passed by userentry.</p>
showkbkeysets	<p>showkbkeysets = 0 – do not display the Knowledge Base Queue Key Set module.                  showkbkeysets = 1 – display the Knowledge Base Queue Key Set module.</p> <p><b>Note:</b> the Queue, Key 1, Key 2 and Key 3 values of the Call will be passed to this module and a list of Knowledge Base articles returned based on these values.</p>
Nocall	<p>nocall = 0 (default) – display versaSRS Call management window (new or existing Call).                  nocall = 1 – do not display the versaSRS Call management window.</p>

### Example Contact Lookup SQL Query

```
SELECT CID AS ContactID, Company AS CField1, FirstName AS CField2, Surname AS CField3, BEmail AS CField4, BPhone AS CField5, BDivision AS CField6, BDepartment AS CField7 FROM tbl_Contacts WHERE CID = [CONTACTLOOKUPKEY]
```

where CField1 ... CField10 map to the contact fields of the Primary Requestor. Note: the query does not need to return all 10 fields.

In the Query above [CONTACTLOOKUPKEY] is replaced with the value defined by *userentry*.

The query may also return the following values defined by the parameters: CompanyID, ContactID, CompanyIDGUID and ContactIDGUID. If the fields are available in the query their values are returned and are recorded against tbl\_Calls.CompanyID, tbl\_Calls.ContactID, tbl\_Calls.CompanyIDGUID and tbl\_Calls.ContactIDGUID respectively. This allows integration with standard versaSRS contacts and versaCRM contacts or any other contact data.

For example, a query could be constructed to return versaCRM contact data based on a unique customer PIN number stored against their contact record.

**IMPORTANT:** The primary requestor contact details defined within the Call template have priority over the requestor's contact details. Therefore, these template fields should remain empty if you wish to populate the Call with the requestor's contact details.

### Special Feature

Some phone systems prefix the value passed as *userentry* with zeros. versaSRS has been designed to remove these leading zeros.



**Example:** 004512 is converted to 4512 before processing

#### **VERY IMPORTANT**

Some phone systems append # character(s) to the value passed as *userentry*. The # character is interpreted by the Web browser as a request to jump to a reference within a Web page. Furthermore, if this parameter is followed by other parameters and values they will be truncated. It is not possible to intercept this condition and modify the URL since the Web browser causes this behavior before the server-side code can perform any processing. The only solution is to make sure that the *userentry* parameter / value is passed at the end of the URL.

**Example:**

<http://myVersaSRS/phonetools.aspx?userentrymode=3&keyPress=551256&callerID=9876&userentry=31481#>

### **3.1 Request To Open An Invalid Call**

In the event that the Caller has requested a Call that does not exist, the Operator will be presented with a single window displaying the message:

*The Call XXX does not exist.*

where XXX is the ticket / Call number requested by the Caller

### **3.2 Request To Open A Call Locked By Another Operator**

In the event that the Caller has requested an existing Call that is currently locked by another Operator, the Operator for the Caller will be presented with a single window displaying the message:

*Call XXX is currently locked by YYY.*

where XXX is the ticket / Call number requested by the Caller and YYY is the name of the Operator that has locked the Call.

### 3.3 Request To Open An Existing Call Where The Operator Is Not A Member Of The Queue

In the event that the Caller has requested an existing Call that is resides within a Queue that the Operator is not a member of, the Operator will be presented with a single window displaying the versaSRS Permission Denied message:

### 3.4 Phone Route Logging

versaSRS has been enhanced to support phone route logging. Details of the requested phone route are recorded in the versaSRS database table, tbl\_PhoneRouteLogs. The database table schema is presented below:

Field	Description
LogID	integer – unique integer value defining the log entry
DateEntered	datetime – the date and time the entry was recorded
UserEntryMode	int– records the value passed by the URL parameter userentrymode
UserEntry	nvarchar(50) – records the value passed by the URL parameter userentry
KeyPress	nvarchar(50) – records the value passed by the URL parameter keypress
CallerID	int – records the value passed by the URL parameter callerid
CallID	int – records the unique versaSRS CallID resulting from the phone route
TemplateID	int – records the unqiue versaSRS Call TemplateID associated with the phone route
ShowContacts	bit – records value of showcontacts assigned to the phone route
ShowKBKeySets	bit – records value of showkbkeysets assigned to the phone route
NoCall	bit – records value of nocall assigned to the phone route

## 4 System Requirements

versaSRS 3.3.0 – Phone System Release  
versaSRS 4.1.0 and above