



PureConnect®

2019 R3

Generated:

11-September-2019

Content last updated:

23-May-2019

See [Change Log](#) for summary of changes.



Custom ODBC Contact Directories

Technical Reference

Abstract

This document describes how to create a Contact tab in the CIC client based on customer data stored in a SQL database or similar repository. It includes data definitions for the `Contacts` table, `SpeedDialList` table, and `SpeedDial` table. It also provides Contact Schema installation scripts for SQL Server and Oracle.

For the latest version of this document, see the PureConnect Documentation Library at: <http://help.genesys.com/cic>.

For copyright and trademark information, see https://help.genesys.com/cic/desktop/copyright_and_trademark_information.htm.

Table of Contents

Table of Contents	2
Introduction to Custom ODBC Contact Directories	3
CIC clients	3
DataManager	3
Contact Data	3
Data Types and Lengths	3
Configuration	3
Speed Dials	3
Data Definitions	4
Contacts table	4
SpeedDialList table	5
SpeedDial table	6
SQL Server Contact Schema Installation Script	7
Oracle Contact Schema Installation Script	12
Change Log	14

Introduction to Custom ODBC Contact Directories

This document describes how to create a Contact tab in the CIC client based on customer data stored in a SQL database or similar repository.

CIC clients

Customer Interaction Center (CIC) supports two interaction management client applications. This documentation uses the term *CIC client* to refer to either Interaction Connect or Interaction Desktop.

DataManager

DataManager is the CIC subsystem that services contact and speed dial requests from the CIC client. Besides the SQL Server contact database that CIC ships with, DataManager can use contact data from multiple sources, including Outlook contacts, Exchange address books, and any ODBC-compliant data source that contains a contact schema equivalent to the base CIC contact schema (often accomplished by views).

You must configure each contact data source for DataManager in Interaction Administrator, both as a generic CIC data source, and as a DataManager data source. In addition, each ODBC-based data source must also be configured in the ODBC setup panel.

Contact Data

The DataManager contact system architecture has limited extensibility. Arrange your contact data table to look almost exactly like the CIC Contacts table. Specifically, give the columns the same names, and compatible types and lengths.

You could set up a view to achieve compatibility. However, a view is not always possible, depending on the schema and data formatting differences. In cases where a view is not possible, you can create another table that you periodically update.

Data Types and Lengths

The data types in your table/view do not have to match the data types in the CIC Contacts table exactly; however, they must be similar types. For example, the data type of the **BusinessCity** field in the Contacts table is `nvarchar`. Your **BusinessCity** field could be `varchar`, `char`, or `nchar` (or any other string type supported by your data provider).

Similarly, the data lengths don't have to match exactly. For example, the declared data length of the **BusinessCity** field in the **Contacts** table is 30 characters (that is, `nvarchar(30)`). Your **BusinessCity** field could safely have a declared length greater than or equal to 30. It might also be able to have a declared length less than 30, but you run the risk of data truncation errors, since the CIC GUIs will not know that shorter limits are in effect.

Configuration

Configure a CIC Data Source for your data provider and source. See [IC Data Source Configuration](#) in *Interaction Administrator* help in the [PureConnect Documentation Library](#).

You also need to configure a Data Manager Contact List Source. Configure it exactly like the IC Public Contacts source (or CIC Private Contacts if you are adding a private directory) except:

- Configure it to use your CIC Data Source.
- Set **TABLE=yourTableOrViewName** entry in the **Additional Information** field (unless your table/view name is also **Contacts**, and you are relying on the qualifier to distinguish your table/view from the CIC table).

Speed Dials

Since speed dial entries can come from multiple data sources, there is little need to have your own SpeedDial and/or SpeedDialList tables. In other words, use CIC tables in most cases. In the rare case where you want to take over the speed dial tables, the data source to use for them can be configured in the main DataManager Configuration page (only one speed dial source is permitted).

Data Definitions

The following is a description of the CIC tables used to store contact data.

Contacts table

Column Name	Type	Length	NULLABLE	Description
ID	nvarchar	25	NOT NULL	This is the unique ID for the row. DataManager uses an algorithm that incorporates a timestamp, a server instance id (usually zero), and an auto-incremented value to generate the ID: <pre>CString DATAMANAGER_DLL DMGetUniqueID(int instanceId) { static long s_nCounter; CTime t = CTime::GetCurrentTime(); long nCounter = InterlockedIncrement(&s_nCounter) % 10000; CString strId; strId.Format(_T("%04d%02d%02d%02d%02d%04d%04d"), (int) (t.GetYear()), (int) (t.GetMonth()), (int) (t.GetDay()), (int) (t.GetHour()), (int) (t.GetMinute()), (int) (t.GetSecond()), nCounter, instanceId); return strId; } // DMGetUniqueID</pre> However, you do not have to use this algorithm; you can use any algorithm you like, as long as it guarantees uniqueness (or at least has a negligible chance of duplicates).
Owner	nvarchar	25	NULL	The creator/owner of the contact. This must be a valid IC user ID (e.g. "TomH").
Access	int		NULL	(AccessFlags for Oracle) - The access flags bit mask value. The flags are: <pre>#define DM_ACCESS_NONE 0x00000000 #define DM_ACCESS_READ 0x00000001 #define DM_ACCESS_INSERT 0x00000002 #define DM_ACCESS_UPDATE 0x00000004 #define DM_ACCESS_DELETE 0x00000008 #define DM_ACCESS_RENAME 0x00000010</pre> So, private contacts will have an Access value of zero, and public contacts will have a non-zero value. Note that DM_ACCESS_RENAME is only applicable to speed dial lists.
Name	nvarchar	64	NULL	Display name for this contact (e.g. "Tom Hynes").
LastName	nvarchar	30	NULL	Last name.
FirstName	nvarchar	30	NULL	First name.
Company	nvarchar	64	NULL	Company name.
Title	nvarchar	32	NULL	Contact's title (e.g. "Software Engineer").
Department	nvarchar	64	NULL	Contact's department name.

BusinessAddress	nvarchar	128	NULL	Contact's business address (e.g. "7601 Interactive Way"); can be multi-lined.
BusinessCity	nvarchar	30	NULL	Contact's business city.
BusinessState	nvarchar	6	NULL	Contact's business state/province.
BusinessZip	nvarchar	15	NULL	Contact's business zip (5 or 9 digit).
BusinessCountry	nvarchar	30	NULL	Contact's business country.
BusinessEMail	nvarchar	128	NULL	Contact's business email address.
Assistant	nvarchar	64	NULL	Contact's business assistant name.
HomeAddress	nvarchar	128	NULL	Contact's home address (e.g. "1 Elm St."); can be multi-lined.
HomeCity	nvarchar	30	NULL	Contact's home city.
HomeState	nvarchar	6	NULL	Contact's home state/province.
HomeZip	nvarchar	15	NULL	Contact's home zip (5 or 9 digit).
HomeCountry	nvarchar	30	NULL	Contact's home country.
HomeEMail	nvarchar	128	NULL	Contact's home email.
BusinessPhone	nvarchar	80	NULL	Contact's primary business phone number ¹ .
BusinessPhone2	nvarchar	80	NULL	Contact's secondary/alternate business phone number.
HomePhone	nvarchar	80	NULL	Contact's primary home phone number.
HomePhone2	nvarchar	80	NULL	Contact's secondary/alternate home phone number.
Mobile	nvarchar	80	NULL	Contact's mobile/cell phone number.
Fax	nvarchar	80	NULL	Contact's fax number.
Pager	nvarchar	80	NULL	Contact's pager number.
AssistantPhone	nvarchar	80	NULL	Contact's assistant's phone number.
URL	nvarchar	255	NULL	Contact's web URL (usually of his/her company).
PrimaryNumber	nvarchar	2	NULL	A one or two-letter code indicating the primary/preferred contact method: <pre> "B" = Business Phone "B2" = Business Phone 2 "H" = Home Phone "H2" = Home Phone 2 "M" = Mobile Phone "F" = Fax "P" = Pager "A" = Assistant Phone </pre>
Notes	nvarchar	2000	NULL	Free-form, multi-lined, comment field.
Password	nvarchar	12	NULL	The Web Services password; no longer used in IC 2.3+ with Tracker licensed.

¹ All phone numbers should be in the Standardized Format (e.g. +13178723000) if Reverse White Page (RWP) lookups will be performed against this source

SpeedDialList table

Column Name	Type	Length	NULLABLE	Description
ID	nvarchar	25	NOT NULL	Unique ID for the row (refer to the ID column of the Contacts table for format information).
Owner	nvarchar	25	NULL	Creator/owner of the row; must be a valid CIC user ID (e.g. "TomH").
Access	int		NULL	(AccessFlags for Oracle) - The access flags bit mask value (refer to the Access column of the Contacts table for legal values).
ListName	nvarchar	80	NULL	Name of the speed dial list.

SpeedDial table

Column Name	Type	Length	NULLABLE	Description
ID	nvarchar	25	NOT NULL	Unique ID for the row (refer to the ID column of the Contacts table for format information).
ListID	nvarchar	25	NOT NULL	ID field of the SpeedDialList entry (refer to the ID column of the Contacts table for format information).
ContactSource	nvarchar	64	NOT NULL	DataManager Contact List Source name to be used when looking up the contact.
ContactID	nvarchar	255	NOT NULL	ID to use when looking up the contact; Since speed dial contacts can come from various data sources (e.g. MAPI, LDAP, etc.), there is no fixed format for this field - it is specific to the source.
SpeedDialNumber	nvarchar	2	NULL	A one or two-letter code indicating which number to use when dialing the contact: " B " = Business Phone " B2 " = Business Phone 2 " H " = Home Phone " H2 " = Home Phone 2 " M " = Mobile Phone " F " = Fax " P " = Pager " A " = Assistant Phone
AdditionalData	nvarchar	128	NULL	Currently not used.

SQL Server Contact Schema Installation Script

```

/*****
 * Desc: Create the table Contacts
 * $Id: //depot/systest/eic/main/products/eic/src/sql/sqlserver/08table/Contacts.TAB#5
 *
 * $Author: Jared.Alford $
 * $Date: 2003/05/07 $
 * $Change: 27477 $
 *****/
/***** Object: Table Contacts *****/
/*****
IF NOT EXISTS (SELECT * FROM sysobjects WHERE id = object_id(N'[dbo].[Contacts]'))

    and OBJECTPROPERTY(id, N'IsUserTable') = 1)
create table Contacts (
    [ID]                nvarchar

(25) NOT NULL,
    Owner              nvarchar

(25) NULL ,
    Access             int

NULL ,
    [Name]             nvarchar

(64) NULL ,
    LastName           nvarchar

(30) NULL ,
    FirstName          nvarchar

(30) NULL ,
    Company            nvarchar

(64) NULL ,
    Title              nvarchar

(32) NULL ,
    Department         nvarchar

(64) NULL ,
    BusinessAddress    nvarchar (128) NULL

,
    BusinessCity       nvarchar

(30) NULL ,
    BusinessState      nvarchar

(6) NULL ,
    BusinessZip        nvarchar

(15) NULL ,
    BusinessCountry    nvarchar (30) NULL

,
    BusinessEMail      nvarchar

(128) NULL ,
    Assistant          nvarchar

(64) NULL ,
    HomeAddress        nvarchar

(128) NULL ,

```

```

        HomeCity          nvarchar
(30) NULL ,
        HomeState         nvarchar
(6) NULL ,
        HomeZip           nvarchar
(15) NULL ,
        HomeCountry       nvarchar
(30) NULL ,
        HomeEMail         nvarchar
(128) NULL ,
        BusinessPhone     nvarchar
(80) NULL ,
        BusinessPhone2    nvarchar (80)
NULL ,
        HomePhone         nvarchar
(80) NULL ,
        HomePhone2        nvarchar
(80) NULL ,
        Mobile            nvarchar
(80) NULL ,
        Fax               nvarchar
(80) NULL ,
        Pager             nvarchar
(80) NULL ,
        AssistantPhone    nvarchar (80)
NULL ,
        URL               nvarchar
(255) NULL ,
        PrimaryNumber     nvarchar
(2) NULL ,
        Notes             nvarchar
(2000) NULL ,
        Password          nvarchar
(12) NULL
) on [PRIMARY]
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))
    and OBJECTPROPERTY(id, N'IsProcedure') = 1)
    EXEC spinin_update_cksum 'IC', 'Running'
GO
--
IF OBJECTPROPERTY (object_id(N'[dbo].[Contacts]') , 'TableHasPrimaryKey'
) = 0
    ALTER TABLE [dbo].[Contacts]
    ADD CONSTRAINT [PK_Contacts] PRIMARY
    KEY CLUSTERED ([ID])
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))
    and OBJECTPROPERTY(id, N'IsProcedure') = 1)

```



```

EXEC spinin_update_cksum 'IC', 'Running'
GO
--
/*****
* Desc: Create the table SpeedDialList
* $Id: //depot/systest/eic/main/products/eic/src/sql/sqlserver/08table/SpeedDialList.TAB#5
$
* $Author: Jared.Alford $
* $Date: 2003/05/07 $
* $Change: 27477 $
*****/
/***** Object: Table SpeedDialList *****/
/*****
IF NOT EXISTS (SELECT * FROM sysobjects WHERE id = object_id(N'[dbo].[SpeedDialList]'))

    and OBJECTPROPERTY(id, N'IsUserTable') = 1)
create table SpeedDialList (
    [ID]          nvarchar

(25) NOT NULL ,
    Owner        nvarchar (25)

NULL ,
    Access       int NULL ,
    ListName     nvarchar (80) NULL
) on [PRIMARY]
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))

    and OBJECTPROPERTY(id, N'IsProcedure') = 1)
EXEC spinin_update_cksum 'IC', 'Running'
GO
--
IF OBJECTPROPERTY (object_id(N'[dbo].[SpeedDialList]') , 'TableHasPrimaryKey'

) = 0
    ALTER TABLE [dbo].[SpeedDialList]
    ADD CONSTRAINT [PK_SpeedDialList] PRIMARY

KEY CLUSTERED ([ID])
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))

    and OBJECTPROPERTY(id, N'IsProcedure') = 1)
EXEC spinin_update_cksum 'IC', 'Running'
GO
--
/*****
* Desc: Create the table SpeedDial
* $Id: //depot/systest/eic/main/products/eic/src/sql/sqlserver/08table/SpeedDial.TAB#5
$
* $Author: Jared.Alford $
* $Date: 2003/05/07 $
* $Change: 27477 $
*****/
/***** Object: Table SpeedDial *****/
/*****
IF NOT EXISTS (SELECT * FROM sysobjects WHERE id = object_id(N'[dbo].[SpeedDial]'))

    and OBJECTPROPERTY(id, N'IsUserTable') = 1)
create table SpeedDial (
    [ID]          nvarchar

(25) NOT NULL ,
    ListID        nvarchar

```

```

(25) NOT NULL ,
    ContactSource      nvarchar

(64) NOT NULL ,
    ContactID          nvarchar

(255) NOT NULL ,
    SpeedDialNumber    nvarchar (2)

NULL ,
    AdditionalData     nvarchar

(128) NULL
) on [PRIMARY]
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))

    and OBJECTPROPERTY(id, N'IsProcedure') = 1)
    EXEC spinin_update_cksum 'IC', 'Running'
GO
--
IF OBJECTPROPERTY (object_id(N'[dbo].[SpeedDial]') , 'TableHasPrimaryKey'

) = 0
    ALTER TABLE [dbo].[SpeedDial]
    ADD CONSTRAINT [PK_SpeedDial] PRIMARY

KEY CLUSTERED ([ID])
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))

    and OBJECTPROPERTY(id, N'IsProcedure') = 1)
    EXEC spinin_update_cksum 'IC', 'Running'
GO
--
---- remove any system named foreign key on ListID ----
DECLARE @FKname VARCHAR(255),
        @SQLstring VARCHAR(255)
SELECT @FKname = so.name
    FROM sysobjects so
    INNER JOIN syscolumns sc ON sc.id = so.parent_obj
    INNER JOIN sysforeignkeys sfk ON so.id = sfk.constid
    AND sc.colid = sfk.fkey
    INNER JOIN syscolumns scref ON scref.id = sfk.rkeyid
    WHERE so.xtype = 'F'
    AND so.parent_obj = object_id('SpeedDial')
    AND sc.name = 'ListID'
    AND sfk.rkeyid = object_id('SpeedDialList')
    AND scref.name = 'ID'

SELECT @SQLstring = 'ALTER TABLE [dbo].[SpeedDial] DROP CONSTRAINT ' +

    ISNULL(@FKname, '')
IF @FKname IS NOT NULL
    EXEC (@SQLstring)
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))

    and OBJECTPROPERTY(id, N'IsProcedure') = 1)
    EXEC spinin_update_cksum 'IC', 'Running'
GO
--
---- use explicit foreign key names from now on!!! ----
IF EXISTS (SELECT * FROM sysobjects WHERE id = object_id(N'[dbo].[FK_SpeedDial_ListID]'))

    and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
    ALTER TABLE [dbo].[SpeedDial]
    DROP CONSTRAINT [FK_SpeedDial_ListID]
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects

```

```

        WHERE id = object_id(N'[dbo].[spinin_update_cksum]')

and OBJECTPROPERTY(id, N'IsProcedure') = 1)
EXEC spinin_update_cksum 'IC', 'Running'
GO
--
IF NOT EXISTS (SELECT * FROM sysobjects WHERE id = object_id(N'[dbo].[FK_SpeedDial_ListID]'))
and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
    ALTER TABLE [dbo].[SpeedDial] WITH

    NOCHECK
        ADD CONSTRAINT [FK_SpeedDial_ListID]

    FOREIGN KEY ([ListID])
        REFERENCES [dbo].[SpeedDialList] ([ID])
IF (@@ERROR = 0) and EXISTS (SELECT * FROM sysobjects
    WHERE id = object_id(N'[dbo].[spinin_update_cksum]'))

and OBJECTPROPERTY(id, N'IsProcedure') = 1)
EXEC spinin_update_cksum 'IC', 'Running'
GO

```

Oracle Contact Schema Installation Script

```
/******
* Desc: create tables for IC(reporting tables)
* $Id: //depot/systest/eic/main/products/eic/src/sql/oracle/08Table/IC_table.sql#12
$
* $Author: elaine.fang $
* $Date: 2004/04/16 $
* $Change: 72603 $
*****/
define xTblTsp = &1
define xIdxTsp = &2
define xProduct = 'IC'
define xStatus = 'RUNNING'
define xNulls = NULL
set verify off
BEGIN
-- Contacts
crtable('Contacts',
' (
  ID varchar2 (25) NOT NULL,
  Owner varchar2 (25) NULL ,
  AccessFlags number (10) NULL ,
  Name varchar2 (64) NULL ,
  LastName varchar2 (30) NULL ,
  FirstName varchar2 (30) NULL ,
  Company varchar2 (64) NULL ,
  Title varchar2 (32) NULL ,
  Department varchar2 (64) NULL ,
  BusinessAddress varchar2 (128) NULL ,
  BusinessCity varchar2 (30) NULL ,
  BusinessState varchar2 (6) NULL ,
  BusinessZip varchar2 (15) NULL ,
  BusinessCountry varchar2 (30) NULL ,
  BusinessEMail varchar2 (128) NULL ,
  Assistant varchar2 (64) NULL ,
  HomeAddress varchar2 (128) NULL ,
  HomeCity varchar2 (30) NULL ,
  HomeState varchar2 (6) NULL ,
  HomeZip varchar2 (15) NULL ,
  HomeCountry varchar2 (30) NULL ,
  HomeEMail varchar2 (128) NULL ,
  BusinessPhone varchar2 (80) NULL ,
  BusinessPhone2 varchar2 (80) NULL ,
  HomePhone varchar2 (80) NULL ,
  HomePhone2 varchar2 (80) NULL ,
  Mobile varchar2 (80) NULL ,
  Fax varchar2 (80) NULL ,
  Pager varchar2 (80) NULL ,
  AssistantPhone varchar2 (80) NULL ,
  URL varchar2 (255) NULL ,
  PrimaryNumber varchar2 (2) NULL ,
  Notes varchar2 (2000) NULL ,
  Password varchar2 (12) NULL
) tablespace &xTblTsp',
'&xProduct',
'&xStatus');
crPK ('CONTACTS', 'PK_CONTACTS', 'ID', '&xProduct', '&xStatus');
-- SpeedDialList
crtable('SpeedDialList',
' (
  ID varchar2 (25) NOT NULL,
  Owner varchar2 (25) NULL ,
  AccessFlags number (10) NULL ,
  ListName varchar2 (80) NULL
) tablespace &xTblTsp',
'&xProduct',
'&xStatus');
```

```

crPK ('SPEEDDIALLIST', 'PK_SPEEDDIALLIST', 'ID', '&xProduct', '&xStatus');
-- SpeedDial
crtable('SpeedDial',
' (
  ID varchar2 (25) NOT NULL ,
  ListID varchar2 (25) NOT NULL REFERENCES SpeedDialList(ID),
  ContactSource varchar2 (64) NOT NULL ,
  ContactID varchar2 (255) NOT NULL ,
  SpeedDialNumber varchar2 (2) NULL ,
  AdditionalData varchar2 (128) NULL
) tablespace &xTblTsp',
'&xProduct',
'&xStatus');
crPK ('SPEEDDIAL', 'PK_SPEEDDIAL', 'ID', '&xProduct', '&xStatus');

```

Change Log

The following table lists the changes to this document since Customer Interaction Center version 4.0 product availability.

Date	Change
01-August-2014	Updated documentation to reflect changes required in the transition from version 4.0 SU# to CIC 2015 R1, such as updates to product version numbers, system requirements, installation procedures, references to Product Information site URLs, and copyright and trademark information.
01-July-2015	Updated cover page with new logo.
01-September-2015	Updated documentation to reflect the addition of two CIC client applications, Interaction Desktop and Interaction Connect and the removal of Interaction Client .NET Edition.
01-April-2017	Updated documentation to reflect the removal of Interaction Client Web Edition.
23-March-2018	Rebranded to Genesys.
23-May-2019	Reorganized the content only, which included combining some topics and deleting others. For more details, see CICDOC-192 .