PureConnect®

2018 R5

Generated:

12-November-2018

Content last updated:

22-March-2018

See Change Log for summary of changes.

Using LDAP for CIC Contact Lists

Technical Reference

Abstract

This document discusses how to setup an LDAP directory to be a data source for an IC Server and includes examples.

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

Table of Contents

About LDAP

Lightweight directory access protocol (LDAP) is an Internet standard that allows applications to connect to a directory server. The LDAP-compatible directory server stores objects that represent users, groups of users, workstations, applications, distribution lists, and so on. Each object has its own set of unique attributes. A user object might have attributes like full name, password, telephone number, and so on. The formal definition of all object types that can be stored in the directory is called a schema. The definition in the schema consists of a class definition for each type of object and an attribute definition for each attribute. The class definitions list the attributes that can be used to describe the class.

From the user interface of the CIC client, a user object is referred to as a contact. Contacts are stored in schema-based lists called contact directories. A contact directory can be either public—available to all CIC client users, or private—available only to the CIC client user who created or owns that directory. LDAP-based enterprise directories can serve as a source for public and private directories. This is useful if you have existing LDAP contact data, or an LDAP directory server and you don't want the additional expense of an SQL server or Oracle license.

Get started

This document assumes that you have installed Customer Interaction Center and Active Directory Server (hereinafter referred to as Active Directory Server), and that each is a member of the same local area network.

About LDAP vendor and version support

Genesys does not officially certify compatibility with LDAP vendors and versions. Instead, selected vendors and versions are periodically target for testing (this is consistent with Genesys stance on RDBMS vendors and versions).

This does not mean that Genesys cannot work with other vendors and versions. For example, there are customers successfully using Active Directory's LDAP implementation. Basically, if an LDAP vendor's server and client APIs conform to the LDAP standard, those should have normal data access.

This document also assumes the following:

- You understand the relationship between a data source, a contact list source, and the contact directory available from the CIC client.
- You are familiar with the features of Interaction Administrator's CIC Data Sources and Contact List Sources.
- You are aware of the concepts of directory services and have a basic understanding of LDAP v-3 protocol.

For a partial list of definitions for the terminology used in this book, see Glossary.

Method for setting up LDAP contact directories

This document describes how to set up contact directories that use the LDAP-enabled Active Directory Server. This support is performed by making use of an existing LDAP schema. With this method, no modification to the schema is necessary; however, not all Customer Interaction Center (CIC) contact features are available.

Use an existing LDAP schema

You can configure CIC to use an existing LDAP schema. This method limits the availability of some CIC contact features. For example, user/person object classes such as `inetorgperson` do not provide contact fields such as home address or business phone2. This method also limits access control so that you cannot specify who can modify a contact. However, using an existing LDAP schema does not require installation of any additional schema objects.

The following shows the process for using an existing LDAP schema. The only requirement is to configure Interaction Administrator.
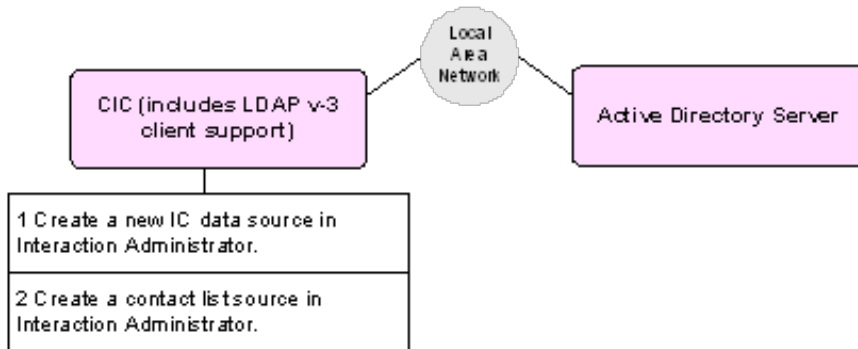
Create a CIC data source
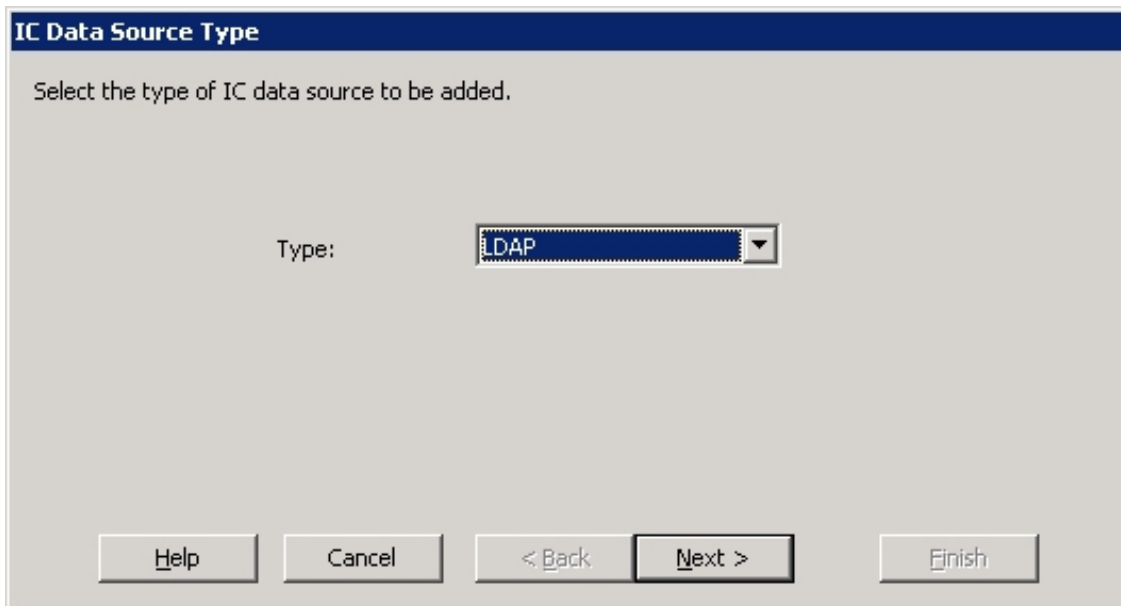
Follow these steps to create and configure an CIC LDAP-compatible data source:

1. From Interaction Administrator, select ⊞ **IC Data Sources**, right-click in the **IC Data Source Name** area and select **New**. The **Entry Name** dialog appears.
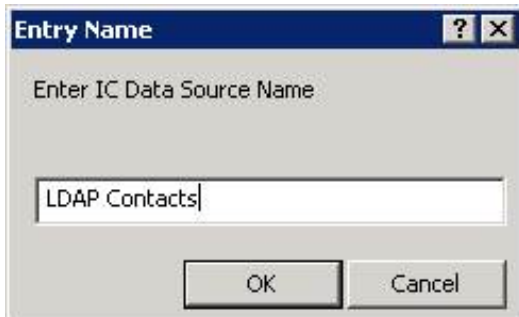
> **Entry Name** [?] [X]
>
> Enter IC Data Source Name
>
> LDAP Contacts|
>
> [OK] [Cancel]

2. Type the name of the new CIC data source. You can name it whatever you like. For this example, it is called `LDAP contacts`.
3. Click **OK**.
   The **IC Data Source Type** dialog appears.

> **IC Data Source Type**
>
> Select the type of IC data source to be added.
>
> Type: [LDAP ▼]
>
> [Help] [Cancel] [< Back] [Next >] [Finish]

4. From the **Type** list, select **LDAP**, then click **Next**.
   The **LDAP Data Source Configuration** window appears.

LDAP Data Source Configuration

Subtype: `Active Directory` ▼ ☑ Read Only

Connection Information

Host Name: [_____] Port: [_____]

Bind DN: [_____] Password: [_____]

Search DN: [_____]

Additional Information: [_____]

[Help] [Cancel] [< Back] [Next >] [Finish]

5. Use the following to fill in the fields as appropriate.

Subtype

Select **Active Directory**. Currently, support is limited to Netscape Directory Server.

Read Only

Select this option if you do not want users to modify contact information.

Host Name

Type the name of the Active Directory Server.

Port

Type the port number used by the Active Directory Server. If blank, then the default 389 is used. Port 3268 can be used here if the connecting to a Global Catalog instead of a specific domain controller.

Bind DN

Specify a *<domain>\<userid>* string value instead of a typical DN. For example: `/b="AiriusDomain\ic_admin"`

Password

The password used to bind with.

Search DN

The Distinguished Name of the node where searching begins. To define the search scope and other search options, please see the Additional Information section below.

Additional Information

Currently, there is no additional information defined for CIC LDAP data sources.

The following shows the minimum settings for an LDAP data source configuration.

6. Click **Finish** to complete the setup.

Create a contact list source

Now that you have created the CIC LDAP-compatible data source, the next task is to create a contact list source that will use the data source. Follow these steps:

1. From Interaction Administrator, expand the **Contact Data Manager** node and select **Contact List Sources**.
2. Right-click in the **Contact List Source Name** area and select **New** from the menu that appears.
   The **Entry Name** dialog opens.
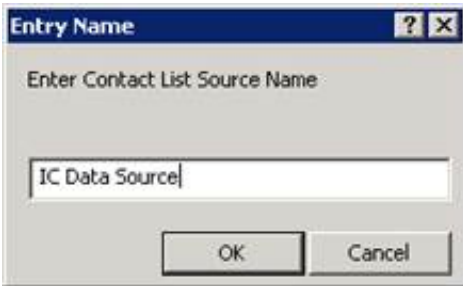
3. Type the name of the new contact list source. This will become the name that appears on the CIC client's Directory tab. Click **OK**.
   The **Contact List Source Configuration** window appears.
4. From the **IC Data Source** list, select the name of the data source you just created in Creating a new CIC data source.
5. Select the **Public** option for public contact sources.

   Note: When creating a Private contact, this box is cleared. Refer to Private contact considerations.

6. From the **Driver** list, select **IC LDAP Contacts**.
   The window will be similar to the following:
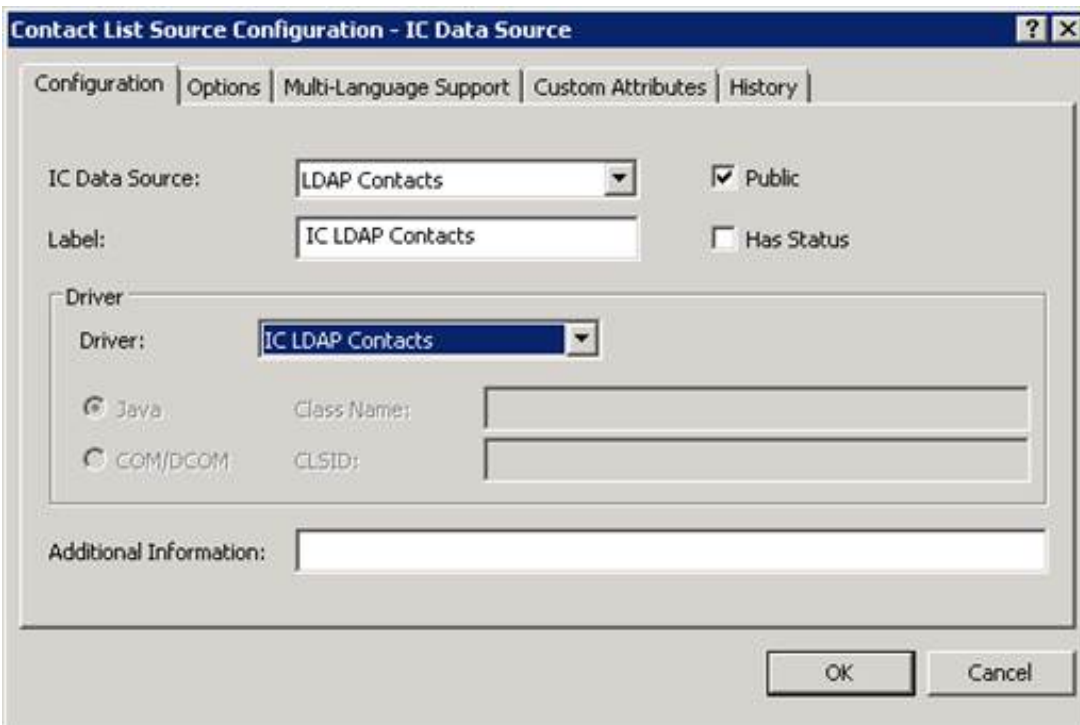
7. In the **Additional Information** box, type configuration information to extend the settings not covered by the above fields. Use the following syntax:
   ```
   <Attribute>=<Value>[;<Attribute>=Value…]
   ```

Configuration syntax	Definition
`Attribute`	Uppercase or lowercase letters are allowed. Case is not preserved.
`Value`	Uppercase or lowercase letters are allowed. Case is preserved.
`=`	Blank spaces around equal sign (=) are allowed. When `Value` contains one or more equal signs (=), then `Value` must be enclosed by double quotation marks (`""`).
`;`	Blank spaces around semi-colon (;) are allowed. When `Value` contains one or more semi-colons (;), then `Value` must be enclosed by double quotation marks(`""`).

Additional Information attributes

| Configuration syntax	Definition
`SEARCH_DN=`	Use to override the search DN specified in the Interaction Administrator CIC Data Source configuration. In LDAP, a search involves three parameters: ○ `SEARCH_DN=` a place (node) in the directory tree to start searching from. ○ `SEARCH_FILTER=` a filter/query to be applied to the contact entries. ○ `SEARCH_SCOPE=` a specification for how/if the search is to proceed down the tree. When specifying the search DN, enclose the string in double quotation marks (`""`), since the standard DN format contains equal signs (=). You can use one or more user substitution variables inside the DN; this is particularly useful when setting up private contacts (see the section on `USERNAME_MAPPING_FILE=`). In addition, you can use one or more built-in substitution variables. They are: `$ICUID` - the CIC user ID. Generally only useful for private contact sources. `$CONTACTSOURCE` - the name of the contact list source used by the Data Manager. For example: Our Public Contacts. `$BINDDN` - The DN used to bind with, as specified in the Interaction Administrator CIC Data Source configuration. **Note:** These variables must all be in uppercase letters.
`SEARCH_FILTER=`	Contains, in LDAP search syntax, the search filter/query to be used. For example: `(&(objectclass=i3person)(i3owner=$ICUID))` You can use one or more user substitution variables inside the DN; this is useful when setting up private contacts (see the section on `USERNAME_MAPPING_FILE`). In addition, you can use one or more built-in substitution variables. They are: ○ `$ICUID` - the CIC user ID. Generally only useful for private contact sources. ○ `$CONTACTSOURCE` - the name of the contact list source used by Data Manager. For example: Our Public Contacts. ○ `$BINDDN` - the DN used to bind with, as specified in the Interaction Administrator CIC Data Source configuration. ○ `$SEARCHDN` - the search DN as specified in the Interaction Administrator CIC Data Source Configuration (**Search DN** field), or as overridden in the contact list source. ○ `$SEARCHDNPARENT` - the parent DN of the configured search DN. **Note:** These variables must all be in uppercase letters.
`SEARCH_SCOPE=`	Specifies how/if the search proceeds down the sub-tree rooted by the search DN. It can have one of three values: ○ `ONE` - This is the default. Searches entries only directly below the search DN. ○ `SUB` - Starts the search at the search DN and then searches everything below, including the search DN. ○ `BASE` - Searches only the search DN entry.

`ATTRIBUTE_MAPPING_FILE=`	By default, CIC uses an object class called `i3person` (which inherits from `inetorgperson`) for storing contact information. However, you are free to choose whatever object class you like, including a custom object class. To do this, you must create a file that contains the mappings between the CIC attributes and the attributes of your object class. CIC provides two sample map files, `I3PERSON_MAP.TXT` and `INETORPERSON_MAP.TXT`, that contain syntax and usage documentation in the files themselves. **Note:** `I3PERSON_MAP.TXT` is provided for informational purposes, and unless you want to modify any of the defaults you do not need to have an `ATTRUBUTE_MAPPING_FILE=` entry for it, since CIC knows about the mappings for the `i3person` object class. For more information, see i3person_map.txt example file and inetorgperson_map.txt.
`USES_FOLDERS=`	The default is **True**. Set to **False** if the contact entries are not rooted by a folder/container entry that you want CIC to create. CIC has a default folder object class called `i3genfolder` that you can use.
`FOLDER_ATTRIBUTE_FILE=`	You can use your own folder type (such as a folder other than `i3genfolder`) and still have CIC automatically create the folders to contain your contacts. To do this you must create a file that contains information about the attributes of the folder object class you wish to use. An example file `I3GENFOLDER_MAP.TXT` is provided whose syntax and usage is the same as the attribute mapping file, and is documented in comments inside the file itself. **Note:** CIC knows about the mappings for `i3genfolder`, so unless you need to modify the defaults, you don't need to have a `FOLDER_ATTRIBUTE_FILE=` entry if you plan on using the default `i3genfolder` object class. For more information on the sample folder attribute file, see i3genfolder_map.txt.
`RDN_ATTRIBUTE=`	Each newly created contact must have a globally unique distinguished name (DN). The left-most component of a distinguished name is called a relative distinguished name (RDN), and contains the attribute-value pair that uniquely defines the contact entry within the current directory. By default, Data Manager will use the `userid` attribute as the attribute for the RDN. However, you can choose whatever attribute you want by setting this option. **Note:** Even though you can specify any attribute as an RDN attribute, DataManager will always set it to a unique timestamp value (for example: `200009231456450001000`. So when choosing an RDN attribute, make sure the semantics of the attribute are such that a unique timestamp value like this will not cause any problems.
`USERNAME_MAPPING_FILE=`	When accessing private contacts, the only information CIC has when distinguishing one user's contact data from another user's contact data is the CIC user ID value. These values might or might not be the same as the LDAP user ID attribute's values. If the values are not the same, set up a file to specify the mapping from the CIC user ID to the LDAP user ID. In fact, you can do this, even if they are the same, in order to specify other substitution values for a user. The `ICUSER_MAP.TXT` example mapping file provides syntax and usage documentation. For more information, see ICUser_map.txt example file.

Example use of attributes

Syntax for public contact source that uses the default folders: `SEARCH_DN="cn=$CONTACTSOURCE, ou=People, o=Airius.com";`
Syntax for private contact source that uses the default folders: `SEARCH_DN="cn=$CONTACTSOURCE, uid=$ICUID, ou=People, o=Airius.com";`

Syntax for private contact source that does not use the default folders:
`USES_FOLDERS=FALSE; SEARCH_DN="uid=$ICUID, ou=People, o=Airius.com";SEARCH_FILTER="`
`(objectclass=i3person)";`

The following shows an example of a minimal contact list configuration for a public contact source that uses the default folders.
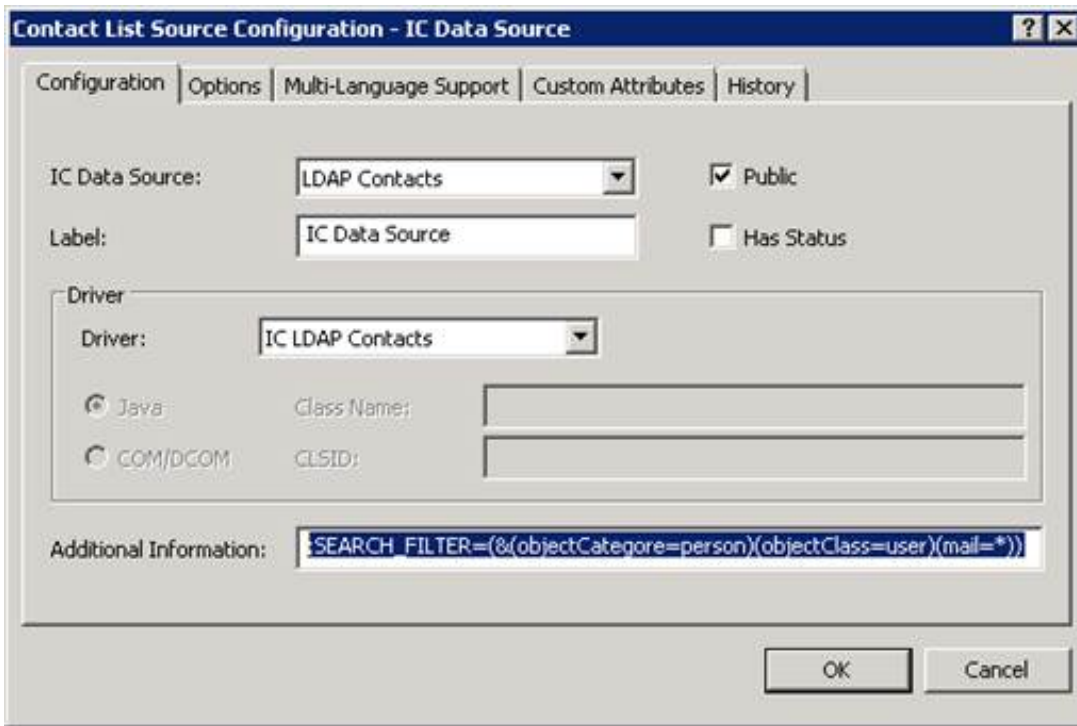
8. When you have set the configuration data, click **OK**.
9. Click the **Options** tab.
10. In the **Timeout (sec)** box, optionally type the timeout in seconds for LDAP operations.
11. In the **Query Row Limit** box, optionally type a value that limits the number of rows to return. This is a safeguard against accidentally loading large numbers of contact entries. The following shows an example of an **Options** tab configuration:
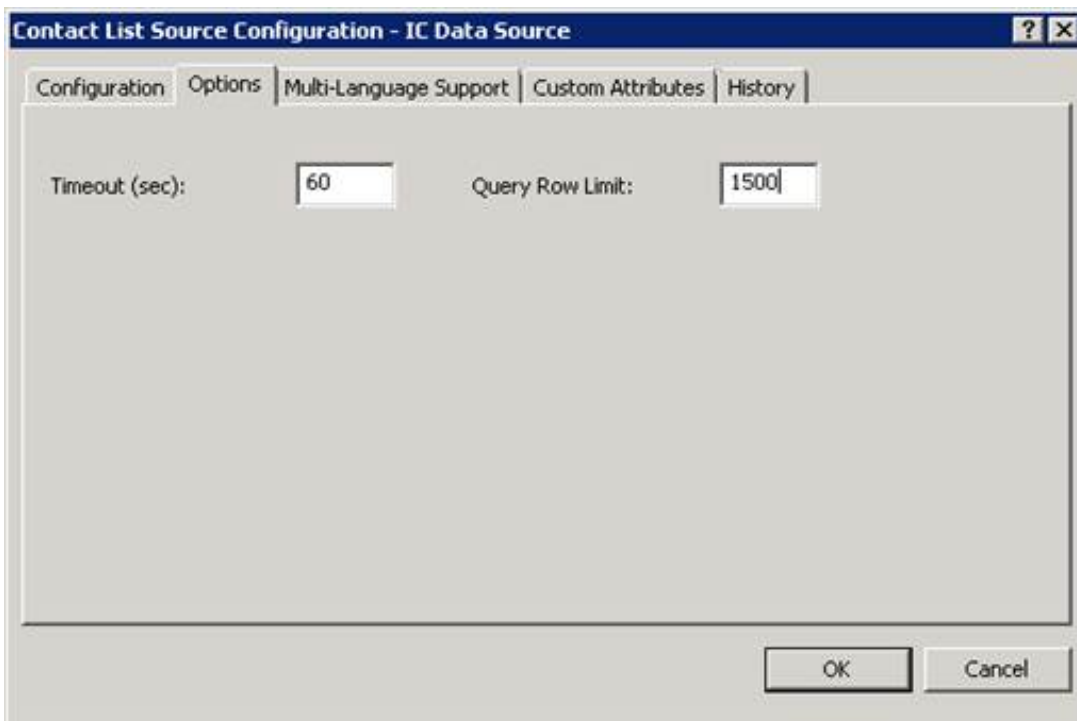
12. Click **OK**.

Private contact considerations

When implementing a private contact list, you should be aware of the following.

User ID values

The only information CIC has when distinguishing one user's private contact data from another user's private contact data is the CIC user's ID value. If the CIC user's ID does not directly correspond to an LDAP attribute (such as `userid`) then a mapping file must be created. For information, see Creating a contact list source `USERNAME_MAPPING_FILE=` attribute.

The following example scenarios show how the link between an CIC user's ID and its corresponding LDAP attribute can be used:

- As a user ID substitution variable in the search DN. The search DN would most likely be the DN of the user's actual LDAP user entry. Create private contacts directly beneath this base.
- As a user ID substitution variable in the search DN. The search DN would most likely be the DN of the user's actual LDAP user entry. Create a folder object beneath this base to contain the contacts.
- As a user ID substitution variable in the search filter. The contacts would be contained in a folder object with other users' private contacts.
- As a user ID substitution variable in the search filter. The contacts would be contained in a folder object with other users' private contacts as well as public contacts.
- As a user ID substitution variable in the search filter. The contacts would be under an organizational unit, like **People**.

Access control

You will want to make private contacts available only to the contact owner. This means you must prevent outsider users from accessing them. Typically, in an LDAP directory, even anonymous users have read access to many of the contact attributes. Therefore, special access control must be established through the use of Netscape Directory Server's access control instructions (ACIs).

If you are using a folder dedicated to a single user's private contacts, then ACI just needs to be established for the folder. Note that this is CIC's default configuration, and CIC will automatically insert the correct ACI values when it creates the folder objects.

If you are not using a folder, or if you're using a folder that contains private contact entries for multiple users, then you must establish ACI in the contact entry instead of the folder entry. The sample attribute mapping file, `i3person_map.txt`, has an example of how to set up ACI for the contacts; the sample folder attribute file, `i3genfolder_map.txt`, has an example of how to set up ACI set up for folders. For more information, see i3person_map.txt example file; inetorgperson_map.txt example file; and i3genfolder_map.txt file.

For more information about ACI's, refer to any authoritative documentation on LDAP.

Other considerations

The following sections contain other items to consider.

LDAP connection caching

DataManager will cache LDAP connections for increased performance. Some command-line parameters (set in the process tree registry entry for DataManager) govern the caching behavior:

- `/LDAPConnCacheSize`. Set this to limit the number of connections in the cache; the default is to use the same number as the number of DataManager threads (set in Interaction Administrator; defaults to 16). To disable connection caching, set this to zero. Example: `DataManagerU.exe /LDAPConncacheSize=10`
- `/LDAPConnCacheIdleTimeout`. When an LDAP connection sits idle in the cache for a period of time, it is disconnected and purged from the cache. By default, they are disconnected after 300 seconds (five minutes) of inactivity. Set this parameter to specify your own idle timeout period (in seconds). You can set it to a very large number, say 1000000000, to effectively disable idle connection flushing. Example: `DataManagerU.exe /LDAPConnCacheIdleTimeout=60`

Reverse White Page lookups

You can configure any public DataManager contact list source to participate in the reverse white page (RWP) lookup sequence. By default, DataManager searches CIC `businessphone`, `businessphone2`, and `homephone` attributes. However, you can configure each contact list source to have its own list of phone attributes to search on. This is specified with the `RWP_LOOKUP_PHONE_TYPES` attribute.

Follow these steps:

1. From Interaction Administrator, expand the **Contact Data Manager** node and select **Contact List Sources**.
2. Right-click in the **Contact List Source Name** area and select **New** from the menu that appears.

    The **Entry Name** dialog opens.
3. Type the name of a new white pages contact list source, and then click **OK**.

    The **Contact List Source Configuration** window appears.
4. From the **IC Data Source** list, select the name of the white pages data source that you previously created.
5. Select the **Public** option for public contact sources, if applicable.
6. From the **Driver** list, select **White Pages**.
7. Type the appropriate attributes in the **Additional Information** box, for example:

    `RWP_LOOKUP_PHONE_TYPES=HomePage,BusinessPhone,BusinessPhone2,Mobile;`

    > Use any combination upper- or lower-case letters for `RWP_LOOKUP_PHONE_TYPE` attribute and phone number type values. Valid phone number types are:
    >
    > - `BusinessPhone`
    > - `BusinessPhone2`
    > - `HomePhone`
    > - `HomePhone2`
    > - `AssistantPhone`
    > - `Mobile`
    > - `Fax`
    > - `Pager`
    >
    > For more information on Reverse White Pages, see *Reverse White Page Lookups Technical Reference* located in **Technical Reference Documents** section of the [CIC Documentation Library](#).

i3person_map.txt example file

The following attribute mapping file is located at https://my.inin.com/support/products/ic40/Pages/Utilities-Downloads.aspx.

```
;****************************************************************************
;* CIC Contact Attribute Mapping File for the "i3person" ObjectClass*
;*                                                                         *
;* Purpose:  To define the mappings between the CIC contact attributes and the *
;*           corresponding LDAP attributes.  In addition, it also allows    *
;*           the (optional) specification of a default value, as well as    *
;*           whether or not the attribute is read-only or read-write.       *
;*                                                                         *
;* Syntax/Notes:                                                           *
;*         1) The basic syntax is:                                         *
;*              <ICAttrib> : [<LDAPAttrib>[,<RoRwFlag>[,<DefaultVal>]]]     *
;*                   - or -                                                 *
;*              NON_IC_ATTRIB_<num> : <LDAPAttrib>,RO[,<DefaultVal>]        *
;*            See 6 note below for extended syntax for ICAttr; see note 7   *
;*            below for information on non-IC attributes.                   *
;*         2) Blank or comment lines are ignored; comment lines start with  *
;*            a semicolon.                                                   *
;*         3) The case of the attributes and the case of the RO/RW flag     *
;*            does not matter.                                              *
;*         4) If an LDAP attribute is not specified (i.e. blank), then it    *
;*            implies that there is no LDAP attribute for the corresponding *
;*            CIC attribute - which will cause the CIC attribute to be       *
;*            ignored/disabled.                                             *
;*         5) The RO/RW flag can be set to:                                 *
;*            a. RO - The value is read-only and will not be updated.       *
;*                    However, if a default value is specified, it will be   *
;*                    used for inserting. Note that for non-CIC attributes,  *
;*                    RO is always used.                                    *
;*            b. RW - The value is read-write and will be updated.  If a     *
;*                    default value is specified, it will be used for        *
;*                    inserting.                                            *
;*            c. (blank) - The default, which is RW for CIC attributes, RO   *
;*                         for non-CIC attributes.                          *
;*         6) If the attribute is multi-valued, and you want the multiple    *
;*            values to be presented together with a separator string       *
;*            between each, then append a [<sep>] to the attribute.         *
;*            For example:  objectclass[;]                                  *
;*            When specifying multiple default values, use this separator    *
;*            between each; for example: top;person;organizationalperson     *
;*            Note that separator strings may have more than one character   *
;*            (e.g. &&, --foo--), and it is up to you to choose a separator *
;*            string that will not happen to occur in the middle of any      *
;*            values.                                                        *
;*         7) You may specify other, non-CIC attributes; this is very useful *
;*            for inserting values for required, non-CIC attributes.  The    *
;*            "CIC" attribute you use is NON_IC_ATTRIB_<num>, where num is    *
;*            the sequential number of the attribute, starting from 1.       *
;*            For example:                                                   *
;*                NON_IC_ATTRIB_1, NON_IC_ATTRIB_2, NON_IC_ATTRIB_3, etc.     *
;*            Note: There cannot be any gaps in the sequence; so if you      *
;*                  were to delete NON_IC_ATTRIB_2, you must rename          *
;*                  NON_IC_ATTRIB_3 to NON_IC_ATTRIB_2 (and so on for all     *
;*                  non-IC attributes greater than 2).                      *
;*         8) You may use substitution variables (defined in a user mapping *
;*            file) for default values.  For example:                       *
;*             OWNER :   i3owner, RO, $USER$                                 *
;*            See the online help in IA for documentation for the user       *
;*            mapping file usage and syntax.                                *
;*            Notes: 1. User-related substitution values are only useful     *
;*                      for private contact sources.                         *
;*                   2. Case is preserved (i.e. you need to match what       *
;*                      was given in the user mapping file).                *
;*         9) You may use the built-in substitution variables for default    *
;*            values.  These are:                                           *
;*                $ICUID   - The CIC user ID (note that this is really only   *
```

```
;*                         useful for private contact sources).          *
;*               $CONTACTSOURCE - The contact (DataManager) data source.  *
;*                         For example: Our Public Contacts              *
;*               $BINDDN  - The DN used to bind with, as specified in the  *
;*                         CIC data source configuration.                *
;*               $SEARCHDN - The search DN as specified in the CIC data source*
;*                         or overridden in the contact list source,       *
;*               $SEARCHDNPARENT - The parent DN of the configured search DN*
;*               $ENTRYDN - The DN of this contact entry.                 *
;*           Note: These must be in UPPER case.                          *
;*                                                                        *
;*************************************************************************
;-------------------------------------------------------------------------
; CIC Attribute              :    LDAP
 Attribute[sep],  RO/RW,  Default Value
;-------------------------------------------------------------------------
  ID                         :    dn, RO
  OWNER                      :    i3owner, RO, $ICUID
  ACCESS                     :    i3access
  NAME                       :    cn
  LASTNAME                   :    sn
  FIRSTNAME                  :    givenname
  HOMEADDRESS                :    i3homeaddress
  HOMECITY                   :    i3homecity
  HOMESTATE                  :    i3homestate
  HOMEZIP                    :    i3homezip
  HOMEPHONE                  :    homephone
  COMPANY                    :    o
  TITLE                      :    title
  DEPARTMENT                 :    departmentnumber
  BUSINESSADDRESS            :    street
  BUSINESSCITY               :    l
  BUSINESSSTATE              :    st
  BUSINESSZIP                :    postalcode
  BUSINESSCOUNTRY            :    i3businesscountry
  BUSINESSEMAIL              :    mail
  ASSISTANT                  :    secretary
  HOMECOUNTRY                :    i3homecountry
  HOMEEMAIL                  :    i3homeemail
  BUSINESSPHONE              :    telephonenumber
  BUSINESSPHONE2             :    i3businessphone2
  HOMEPHONE2                 :    i3homephone2
  MOBILE                     :    mobile
  FAX                        :    facsimiletelephonenumber
  PAGER                      :    pager
  ASSISTANTPHONE             :    i3assistantphone
  URL                        :    labeleduri
  PRIMARYNUMBER              :    i3primarynumber
  NOTES                      :    description
  ;---------------- Other non-CIC, but needed attributes--------------------
  NON_IC_ATTRIB_1            :    objectclass[;], RO, top;Person;organizationalperson;inetorgperson;i3person
  ; By default, both PUBLIC and PRIVATE contact entries are parented by by
  ; a folder (i3genfolder by default), and therefore do not have ACI entries
  ; (i.e. they use the ACI of their parent folder).
  ; This ACI would be suitable for a PRIVATE contact source that has the contact
  ; entries parented directly by the user entry.
  ; NON_IC_ATTRIB_2          :    aci[&&], RO, (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Hide from everyone except EIC and contact owner."; deny (all) userdn !=
"ldap:///$BINDDN||ldap:///$SEARCHDN";) && (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl "Allow
EIC and contact owner all access."; allow (all) userdn = "ldap:///$BINDDN||ldap:///$SEARCHDN";)
  ; This ACI would be suitable for a PUBLIC contact source that has the contact
  ; entries parented by a non-folder entry (e.g. the People ou).
  ; NON_IC_ATTRIB_2          :    aci[&&], RO, (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Allow CIC all access."; allow (all) userdn = "ldap:///$BINDDN";)
```

inetorgperson_map.txt example file

The following attribute mapping file is located at https://my.inin.com/support/products/ic40/Pages/Utilities-Downloads.aspx.

```
;****************************************************************************
;* CIC Contact Attribute Mapping File for the LDAP "inetorgperson" ObjectClass *
;*                                                                          *
;* Purpose:  To define the mappings between the CIC contact attributes and the *
;*           corresponding LDAP attributes.  In addition, it also allows    *
;*           the (optional) specification of a default value, as well as    *
;*           whether or not the attribute is read-only or read-write.       *
;*                                                                          *
;* Syntax/Notes:                                                            *
;*           1) The basic syntax is:                                        *
;*                 <ICAttrib> : [<LDAPAttrib>[,<RoRwFlag>[,<DefaultVal>]]]  *
;*                       - or -                                             *
;*                 NON_IC_ATTRIB_<num> : <LDAPAttrib>,RO[,<DefaultVal>]     *
;*              See 6 note below for extended syntax for ICAttr; see note 7 *
;*              below for information on non-CIC attributes.                *
;*           2) Blank or comment lines are ignored; comment lines start with *
;*              a semicolon.                                                 *
;*           3) The case of the attributes and the case of the RO/RW flag   *
;*              does not matter.                                            *
;*           4) If an LDAP attribute is not specified (i.e. blank), then it *
;*              implies that there is no LDAP attribute for the corresponding *
;*              CIC attribute - which will cause the CIC attribute to be    *
;*              ignored/disabled.                                           *
;*           5) The RO/RW flag can be set to:                               *
;*              a. RO - The value is read-only and will not be updated.     *
;*                      However, if a default value is specified, it will be *
;*                      used for inserting. Note that for non-CIC attributes, *
;*                      RO is always used.                                  *
;*              b. RW - The value is read-write and will be updated.  If a  *
;*                      default value is specified, it will be used for     *
;*                      inserting.                                          *
;*              c. (blank) - The default, which is RW for CIC attributes, RO *
;*                           for non-CIC attributes.                        *
;*           6) If the attribute is multi-valued, and you want the multiple *
;*              values to be presented together with a separator string     *
;*              between each, then append a [<sep>] to the attribute.       *
;*              For example:  objectclass[;]                                *
;*              When specifying multiple default values, use this separator *
;*              between each; for example: top;person;organizationalperson  *
;*              Note that separator strings may have more than one character *
;*              (e.g. &&, --foo--), and it is up to you to choose a separator *
;*              string that will not happen to occur in the middle of any   *
;*              values.                                                     *
;*           7) You may specify other, non-CIC attributes; this is very useful *
;*              for inserting values for required, non-CIC attributes.  The *
;*              "CIC" attribute you use is NON_IC_ATTRIB_<num>, where num is *
;*              the sequential number of the attribute, starting from 1.    *
;*              For example:                                                *
;*                  NON_IC_ATTRIB_1, NON_IC_ATTRIB_2, NON_IC_ATTRIB_3, etc. *
;*              Note: There cannot be any gaps in the sequence; so if you   *
;*                    were to delete NON_IC_ATTRIB_2, you must rename       *
;*                    NON_IC_ATTRIB_3 to NON_IC_ATTRIB_2 (and so on for all *
;*                    non-CIC attributes greater than 2).                   *
;*           8) You may use substitution variables (defined in a user mapping *
;*              file) for default values.  For example:                     *
;*                  OWNER :  i3owner, RO, $USER$                            *
;*              See the online help in IA for documentation for the user    *
;*              mapping file usage and syntax.                              *
;*              Notes: 1. User-related substitution values are only useful  *
;*                        for private contact sources.                      *
;*                     2. Case is preserved (i.e. you need to match what    *
;*                        was given in the user mapping file).              *
;*           9) You may use the built-in substitution variables for default *
;*              values.  These are:                                        *
;*                  $ICUID  - The CIC user ID (note that this is really only *
```

```
;*                      useful for private contact sources).               *
;*                  $CONTACTSOURCE - The contact (DataManager) data source.   *
;*                         For example: Our Public Contacts                *
;*               $BINDDN  - The DN used to bind with, as specified in the   *
;*                         CIC data source configuration.                   *
;*               $SEARCHDN - The search DN as specified in the CIC data source*
;*                         or overridden in the contact list source,       *
;*               $SEARCHDNPARENT - The parent DN of the configured search DN*
;*               $ENTRYDN - The DN of this contact entry.                   *
;*           Note: These must be in UPPER case.                            *
;*                                                                          *
;****************************************************************************
;----------------------------------------------------------------------------
; CIC Attribute           :    LDAP
 Attribute[sep],  RO/RW,  Default Value
;----------------------------------------------------------------------------
  ID                      :    dn, RO
  OWNER                   :
  ACCESS                  :
  NAME                    :    cn
  LASTNAME                :    sn
  FIRSTNAME               :    givenname
  HOMEADDRESS             :
  HOMECITY                :
  HOMESTATE               :
  HOMEZIP                 :
  HOMEPHONE               :    homephone
  COMPANY                 :    o
  TITLE                   :    title
  DEPARTMENT              :    departmentnumber
  BUSINESSADDRESS         :    street
  BUSINESSCITY            :    l
  BUSINESSSTATE           :    st
  BUSINESSZIP             :    postalcode
  BUSINESSCOUNTRY         :
  BUSINESSEMAIL           :    mail
  ASSISTANT               :    secretary
  HOMECOUNTRY             :
  HOMEEMAIL               :
  BUSINESSPHONE           :    telephonenumber
  BUSINESSPHONE2          :
  HOMEPHONE2              :
  MOBILE                  :    mobile
  FAX                     :    facsimiletelephonenumber
  PAGER                   :    pager
  ASSISTANTPHONE          :
  URL                     :    labeleduri
  PRIMARYNUMBER           :
  NOTES                   :    description
    ;---------------- Other non-CIC, but needed attributes
    NON_IC_ATTRIB_1         :    objectclass[;], RO, top;Person;organizationalperson;inetorgperson
    ; By default, both PUBLIC and PRIVATE contact entries are parented by by
  ; a folder (i3genfolder by default), and therefore do not have ACI entries
  ; (i.e. use the ACI of their parent folder).
    ; This ACI would be suitable for a PRIVATE contact source that has the contact
  ; entries parented directly by the user entry.
; NON_IC_ATTRIB_2         :    aci[&&], RO, (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Hide from everyone except EIC and contact owner."; deny (all) userdn !=
"ldap:///$BINDDN||ldap:///$SEARCHDN";) && (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl "Allow
EIC and contact owner all access."; allow (all) userdn = "ldap:///$BINDDN||ldap:///$SEARCHDN";)
  ; This ACI would be suitable for a PUBLIC contact source that has the contact
  ; entries parented by a non-folder entry (e.g. the People ou).
; NON_IC_ATTRIB_2         :    aci[&&], RO, (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Allow EIC all access."; allow (all) userdn = "ldap:///$BINDDN";)
```

i3genfolder_map.txt example file

The following attribute mapping file is located at https://my.inin.com/support/products/ic40/Pages/Utilities-Downloads.aspx.

```
;***************************************************************************
;* CIC Contact Folder Attribute Mapping File for the "i3genfolder" ObjectClass *
;*                                                                         *
;* Purpose:  To define the mappings between the CIC contact folder attributes  *
;*           and the corresponding LDAP attributes.  In addition, it also    *
;*           allows the (optional) specification of a default value, as well  *
;*           as whether or not the attribute is read-only or read-write.      *
;*                                                                         *
;* Syntax/Notes:                                                           *
;*         1) The basic syntax is:                                         *
;*              NON_IC_ATTRIB_<num> : <LDAPAttrib>,RO[,<DefaultVal>]        *
;*            See note 6 below for information on non-CIC attributes.        *
;*         2) Blank or comment lines are ignored; comment lines start with  *
;*            a semicolon.                                                  *
;*         3) The case of the attributes and the case of the RO/RW flag     *
;*            does not matter.                                              *
;*         4) The RO/RW flag is always RO for folder attributes.            *
;*         5) If the attribute is multi-valued, and you want the multiple    *
;*            values to be presented together with a separator string       *
;*            between each, then append a [<sep>] to the attribute.         *
;*            For example:  objectclass[;]                                  *
;*            When specifying multiple default values, use this separator   *
;*            between each; for example: top;person;organizationalperson    *
;*            Note that separator strings may have more than one character  *
;*            (e.g. &&, --foo--), and it is up to you to choose a separator *
;*            string that will not happen to occur in the middle of any     *
;*            values.                                                       *
;*         6) You may specify other, non-CIC attributes; this is very useful *
;*            for inserting values for required, non-CIC attributes.  The    *
;*            "CIC" attribute you use is NON_IC_ATTRIB_<num>, where num is    *
;*            the sequential number of the attribute, starting from 1.      *
;*            For example:                                                  *
;*                NON_IC_ATTRIB_1, NON_IC_ATTRIB_2, NON_IC_ATTRIB_3, etc.    *
;*            Note: There cannot be any gaps in the sequence; so if you      *
;*                  were to delete NON_IC_ATTRIB_2, you must rename          *
;*                  NON_IC_ATTRIB_3 to NON_IC_ATTRIB_2 (and so on for all     *
;*                  non-CIC attributes greater than 2).                     *
;*         7) You may use substitution variables (defined in a user mapping *
;*            file) for default values.  For example:                       *
;*              NON_IC_ATTRIB_1 :   cn, RO, $USER$                          *
;*            See the online help in IA for documentation for the user       *
;*            mapping file usage and syntax.                                *
;*            Notes: 1. User-related substitution values are only useful    *
;*                      for private contact sources.                        *
;*                   2. Case is preserved (i.e. you need to match what       *
;*                      was given in the user mapping file).                *
;*         8) You may use the built-in substitution variables for default   *
;*            values.  These are:                                           *
;*              $ICUID  - The CIC user ID (note that this is really only     *
;*                        useful for private contact sources).             *
;*              $CONTACTSOURCE - The contact (DataManager) data source.      *
;*                        For example: Our Public Contacts                  *
;*              $BINDDN  - The DN used to bind with, as specified in the     *
;*                        CIC data source configuration.                    *
;*              $SEARCHDN - The search DN as specified in the CIC data source*
;*                        or overridden in the contact list source,         *
;*              $SEARCHDNPARENT - The parent DN of the configured search DN*
;*              $ENTRYDN - The DN of this contact entry.                     *
;*            Note: These must be in UPPER case.                            *
;*                                                                         *
;***************************************************************************
;-----------------------------------------------------------------
; CIC Attribute              :    LDAP Attribute[sep],  RO/RW,  Default Value
;-----------------------------------------------------------------
  NON_IC_ATTRIB_1            :    objectclass[;], RO, top;i3genfolder
```

```
   NON_IC_ATTRIB_2             :     cn, RO, $CONTACTSOURCE
    ; Unless ACI must be established in each contact entry (for example, the
   ; folder contains private contact entries for more than one user), then
   ; one of the two ACIs below would be needed.
   ; This ACI would be suitable for a PRIVATE contact source that has the contact
   ; entries parented by this folder, which is parented by the user entry.
; NON_IC_ATTRIB_3             :     aci[&&], RO, (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Hide from everyone except EIC and contact owner."; deny (all) userdn !=
"ldap:///$BINDDN||ldap:///$SEARCHDNPARENT";) && (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Allow EIC and contact owner all access."; allow (all) userdn = "ldap:///$BINDDN||ldap:///$SEARCHDNPARENT";)
    ; This ACI would be suitable for a PUBLIC contact source that has the contact
   ; entries parented by this folder.
; NON_IC_ATTRIB_3             :     aci[&&], RO, (target="ldap:///$ENTRYDN")(targetattr="*")(version 3.0; acl
"Allow CIC all access."; allow (all) userdn = "ldap:///$BINDDN";)
```

ICUser_map.txt example file

The following attribute mapping file is located at https://my.inin.com/support/products/ic40/Pages/Utilities-Downloads.aspx.

```
;******************************************************************************
;* Sample CIC User ID to LDAP User ID Mapping File                         *
;*                                                                         *
;* Purpose:  To define the mappings between the CIC user ID and the LDAP user  *
;*           ID if not the same.  If they are the same, then this file is    *
;*           not needed; also, this file is not needed for public contact    *
;*           sources - only private ones are tied to a user.                 *
;*                                                                         *
;* Syntax/Notes:                                                           *
;*          1) The basic syntax is:                                        *
;*               <ICUID> : <attrib>=<value>[,<attrib>=<value>, ...]        *
;*          2) Blank or comment lines are ignored; comment lines start with  *
;*             a semicolon.                                                 *
;*          3) The case of the CIC user ID does not matter. The case of the  *
;*             attribute - which is used as a substitution variable - is    *
;*             preserved (i.e. case-sensitive); the values case is also     *
;*             preserved.                                                   *
;*          4) You can use an asterisk as a wildcard to match "all other"   *
;*             IDs (see the example below).                                 *
;*          5) You may use the built-in substitution variables for default  *
;*             values.  These are:                                         *
;*               $ICUID - The CIC user ID.                                 *
;*               $CONTACTSOURCE - The contact (DataManager) data source.    *
;*                        For example: Our Public Contacts                  *
;*               $BINDDN  - The DN used to bind with, as specified in the   *
;*                        CIC data source configuration (or overridden in   *
;*                        the contact data source).                         *
;*           $SEARCHDN - The search DN as specified in the CIC data source*
;*                        or overridden in the contact list source,         *
;*               $SEARCHDNPARENT - The parent DN of the configured search DN*
;*               $ENTRYDN - The DN of this contact entry.                   *
;*          Notes:                                                         *
;*             1. These must be in UPPER case.                             *
;*             2. Although these built-in substitution variable are        *
;*                available, few (if any) of them are likely to be useful *
;*                here (i.e. they are mainly useful for the attribute       *
;*                mapping file).                                           *
;******************************************************************************
;----------------------------------------------------------------------------
; CIC User ID     :     Attribute=Value[, Attribute=Value, ...]
;----------------------------------------------------------------------------
DonB              : $UID=DBrown
PamH              : $UID=PHynes
TomH              : $UID=THynes
 *                : $UID=$ICUID
```

Glossary

The following pertains to Data Manager and LDAP-specific terminology used in this document.

Access control instructions (ACI)

Access control in LDAP is a way of expressing and configuring the rights that users have to directory entries. A Netscape Directory Server stores access control Instructions inside the directory itself. Every entry can have a set of rules that define an ACI for that entry.

Access control list (ACL)

Most network security systems operate by allowing selective use of services. An access control list is the usual means by which access to, and denial of, services is controlled. It is simply a list of the services available, each with a list of the computers and users permitted to use the service.

Company Directory

The Company Directory contains the phone number or extension for every person running the CIC client.

Contact

In the CIC client, a contact is information about a person, including the person's name, telephone number, e-mail address, mailing address, web page, and so on. Contacts are stored in lists called contact directories.

Contact directory

Contact directories, also called data sources, contain lists of people and contact information about those people. Add and remove contact directories by right-clicking in the gray area to the right of the directory pages.

A contact directory may be either public (available to all CIC client users) or private (available only to the CIC client user who created or owns that directory).

By default, the contact directory for your private contacts is named **CIC Private Contacts**. The default contact directory for your public contacts is named **CIC Public Contacts**. The contact directory that holds all CIC client users is the Company Directory.

Contact List Source

Contact list sources in Interaction Administrator use CIC data sources and a particular CIC-specific driver to define contact lists that can appear as directory tabs in the CIC client. The **CIC LDAP Contact** driver provides access to the contact data stored in an LDAP-enabled Netscape Directory Server.

Distinguished name (DN)

Distinguished name. Every addressable directory object has a distinguished name (DN) consisting of its path plus its name, applications can address a message to any object using the object's DN.

CIC Data Source

An CIC Data Source is configured in Interaction Administrator, and contains the information needed to connect to the underlying data repository (data source).

LDAP

Lightweight Directory Access Protocol.LDAP is an Internet standard that allows applications to connect to a directory server. The LDAP-compatible directory server stores objects that represent users, groups of users, workstations, applications, distribution lists, and so on.

Private contact

Private contacts are available only to you, and other CIC client users will not see them.

Public contact

Public contacts are available for all CIC client users to see.

Schema

A schema is the formal definition of all object types that can be stored in the directory. The definition in the schema consists of a class definition for each type of object and an *attribute* definition for each attribute. The class definitions list the attributes that can be used to describe the class.

User substitution variables

Substitution variables are placeholders used to denote a string value that will be substituted at a later time (such as runtime). For example, $CONTACTSOURCE is a substitution variable that can be used when configuring a contact list source in Interaction Administrator. At runtime – when the contact data is actually being retrieved – the actual name of the current contact list source is substituted for $CONTACTLISTSOURCE.

A *user* substitution variable is a substitution variable that has different runtime mappings for different users (see ICUser_map.txt example file).

Sources:

Newton, Harry, Newton's Telecom Dictionary, Telecom Books, February 2000.

Weltman, Rob, and Dahbura, Tony, LDAP Programming with Java, AWL Direct Sales, Addison Wesley Longman, Inc, January 2000.

Change Log

Date	Changes
16-September-2015	Updated documentation to reflect logo rebranding and the addition of two CIC client applications, Interaction Desktop and Interaction Connect.
22-March-2018	Rebranded to Genesys.

Change Log

Copyright and Trademark Information