



PureConnect®

2018 R5

Generated:

12-November-2018

Content last updated:

10-May-2018

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



Reverse White Pages

Technical Reference

Abstract

Reverse White Page Lookup refers to the process of using an interaction address, such as a telephone number, to locate the contact information of a party in a database or other data source. The result of a Reverse White Page (RWP) lookup is displayed by the CIC client to identify the caller, by Interaction Tracker to mark the tracked database records with this caller information, and can be used by other applications to pop a customer record when an inbound call arrives. This content presents the RWP process, and explains how to configure CIC to search multiple RWP data sources.

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	3
CIC clients	4
RWP Processing in CIC	5
RWP Data Sources	5
First-Level Sources	5
Secondary Data Sources	6
RWP Processing Diagram	7
The RWP Process	7
Subsystems and Components used when lookups are performed	8
RWP Processing Algorithms	8
Step 1: Initial setting of EIC_RemoteName in CReverseWhitePagesU DLL	9
Step 2: Asynchronous RWP lookup invocation in CReverseWhitePagesU DLL	9
Step 3: RWP lookups in DataManagerU.exe	10
RWP Processing in Interaction Tracker	10
Replicating the candidate result	11
Completing the RWP processing	12
Definitive RWP Results	12
If Tracker is Licensed	12
If Tracker is not Licensed	14
RWP Tools in Interaction Designer	14
WhitePagesLocality Tool	16
WhitePages Tool	16
WhitePagesAsync Tool	19
Configuring Data Sources in Interaction Administrator	22
Install and Register COM Drivers	22
Configure Data Sources for new Drivers	22
Configure an IC Data Source	22
Configure a DataManager Contact List Source	24
Add the source to the RWP Lookup Sequence	27
Developing Drivers for RWP Data Sources	28
IDL file for the IRwp interface - C Interface Definition	28
IDL file for the IRwp interface - C# Interface Definition	29
Initialize Method	29
Lookup Method	29
Deinitialize Method	30
Appendix A: RWP XML Attributes	31
Attributes defined by PureConnect	31
Control and interaction-specific attributes (3rd-pary drivers typically do not set these):	31
Individual attributes:	31
Location attributes (only applicable if Tracker is licensed):	31
Organization attributes (only applicable if Tracker is licensed):	31
Common Individual, Location, and Organization attributes:	32
Individual address and connection attributes:	32
Location address and connection attributes (only applicable if Tracker is licensed):	32
Organization address and connection attributes (only applicable if Tracker is licensed):	32
Driver-Defined Attributes	33
Appendix B: Display Name Generation	34
Selecting RWP display name format in Interaction Administrator.	34
Appendix C: Text File Formats	35
CountryCodes.txt file	35
CountryCodes.txt File Format	35
WhitePages.txt file	36
Whitepages.txt File Format	36
Appendix D: Tracker Contact Resolution Prompt	39
Change Log	40

Introduction

Reverse White Page lookup refers to the process of searching for an interaction address, such as a telephone number, in a database or other data source, and retrieving the contact information (name, street address, etc.) of a caller or party called. The reverse matching process eliminates the need for a call center to capture customer name and locality information, since this information can be retrieved later from a database. Throughout this document, the term Reverse White Page is abbreviated RWP, Interaction Tracker is abbreviated Tracker, and Customer Interaction Center is abbreviated CIC.

For inbound interactions, the interaction address used to look up the contact information is obtained differently depending on the interaction type:

- **Calls, faxes**—the phone number is extracted from the ANI information.
- **Emails**—the sender's email address is taken from the "From" field of the email.

Note: only ACD emails trigger RWP processing.

- **Web chats**—if a DNS lookup on the IP address succeeds, the host name is used; otherwise, the IP address is used.

The result of the Reverse White Page (RWP) lookup can be attached to one or more call object attributes. These attributes are displayed by the CIC client to identify the caller, and can be used by third-party applications to pop specific customer records in response to incoming calls. In CIC 3.0 and later, private RWP information is stored in a call attribute named Eic_TrackerRWPPriateInfo. Public RWP information is stored in an attribute named Eic_TrackerRWPInfo. For more information about attributes, see the *Interaction Attributes Technical Reference* in the PureConnect Documentation Library.

In addition, Interaction Tracker ([if licensed](#)) is notified of the results so that the interaction participant data records get updated with the correct contact information, and so that the Contact Resolution Dialog on the client can decide whether or not it needs to prompt.

Note: Interaction Tracker also has provisions for explicitly tagging an interaction with the correct contact information, independent of the RWP process.

Customer Interaction Center can be configured to search contact data stored in a variety of repositories, including SQL Server and Oracle databases, simple text files, white pages directories, and mail servers such as Exchange. These contact data sources are defined in Interaction Administrator. For details, see "Overview of IC Data Sources and Contact Lists" in the Interaction Administrator help system.

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.

RWP Processing in CIC

This section describes the data sources, subsystems, components, and algorithms used when the Customer Interaction Center server performs RWP lookups.

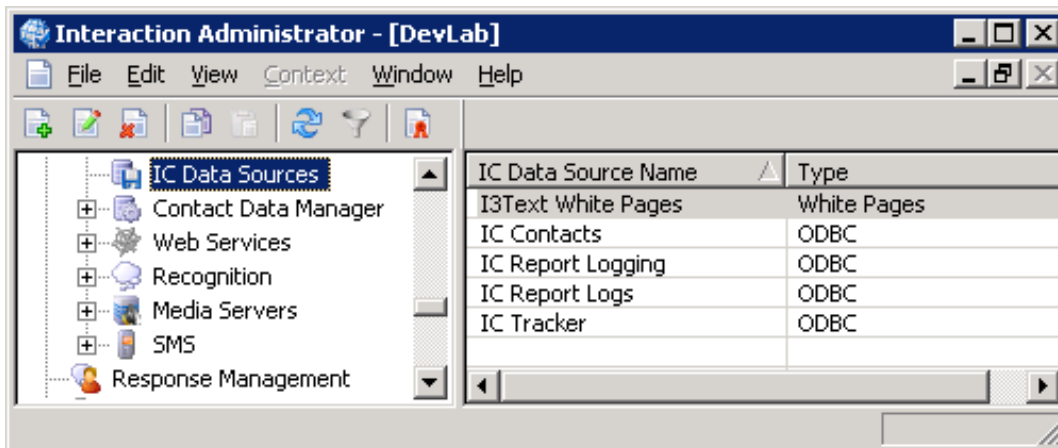
RWP Data Sources

CIC can access contact data stored in a variety of repositories, including SQL Server and Oracle databases. CIC can also read data from simple text files, from white pages databases, LDAP directories, or from directories used by mail servers such as Exchange.

First-Level Sources

CIC uses a two-level hierarchy of data source configuration. First-level sources are configured under the IC Data Sources container in Interaction Administrator. This configuration contains low-level, data source-specific information - such as connection information - that is independent of any CIC application (or subsystem) that may need to connect to that data source.

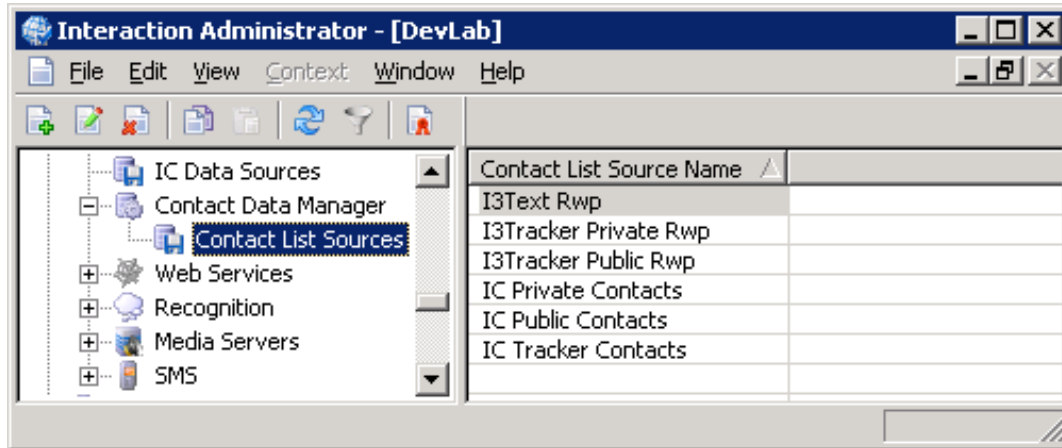
By separating out the low-level data source configuration from any higher-level application source configuration, the low-level data source configuration can be reused by multiple applications without duplication.



Secondary Data Sources

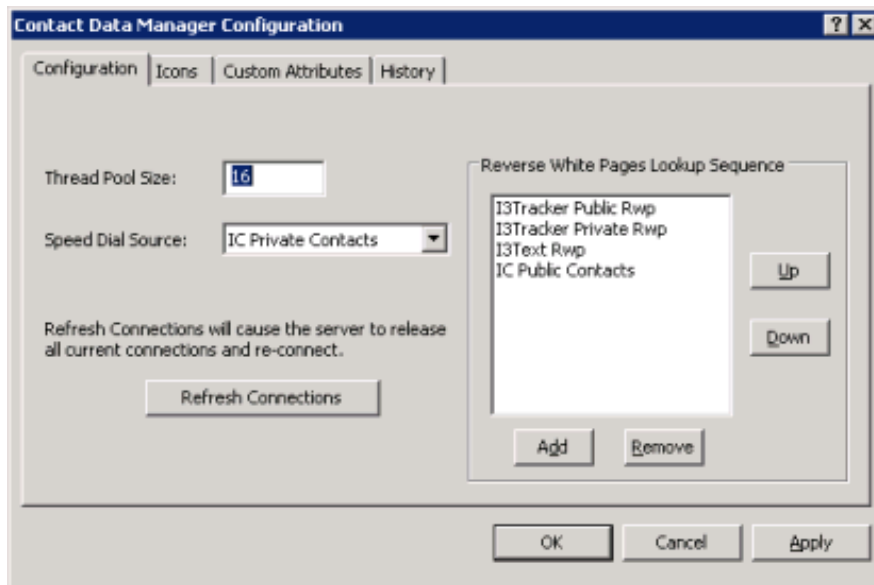
Second-level sources have a link to a first-level data source, and contain application-specific information related to the data source. Second-level sources are configured with their respective applications in IA. For example, Data Manager is configured under the Contact Data Manager container, and Data Manager's second-level sources are configured under the Contact List Sources sub-container.

Since Data Manager is the back-end subsystem that performs the RWP lookups, an RWP data source is actually just a Data Manager Contact List Source that is used for RWP lookups.



Contact List Sources sub-container of the Contact Data Manager container

CIC can be configured to search multiple RWP sources. The order in which the sources are searched is also configurable. In general, the first match terminates the search; however, if Tracker is licensed the search may continue on to gather all matches, depending on how the data sources are configured.



Reverse White Page Lookup Sequence in the Contact Data Manager configuration page

Additional RWP sources, such as Redi-Data's white pages service, can be defined as needed.

RWP Processing Diagram

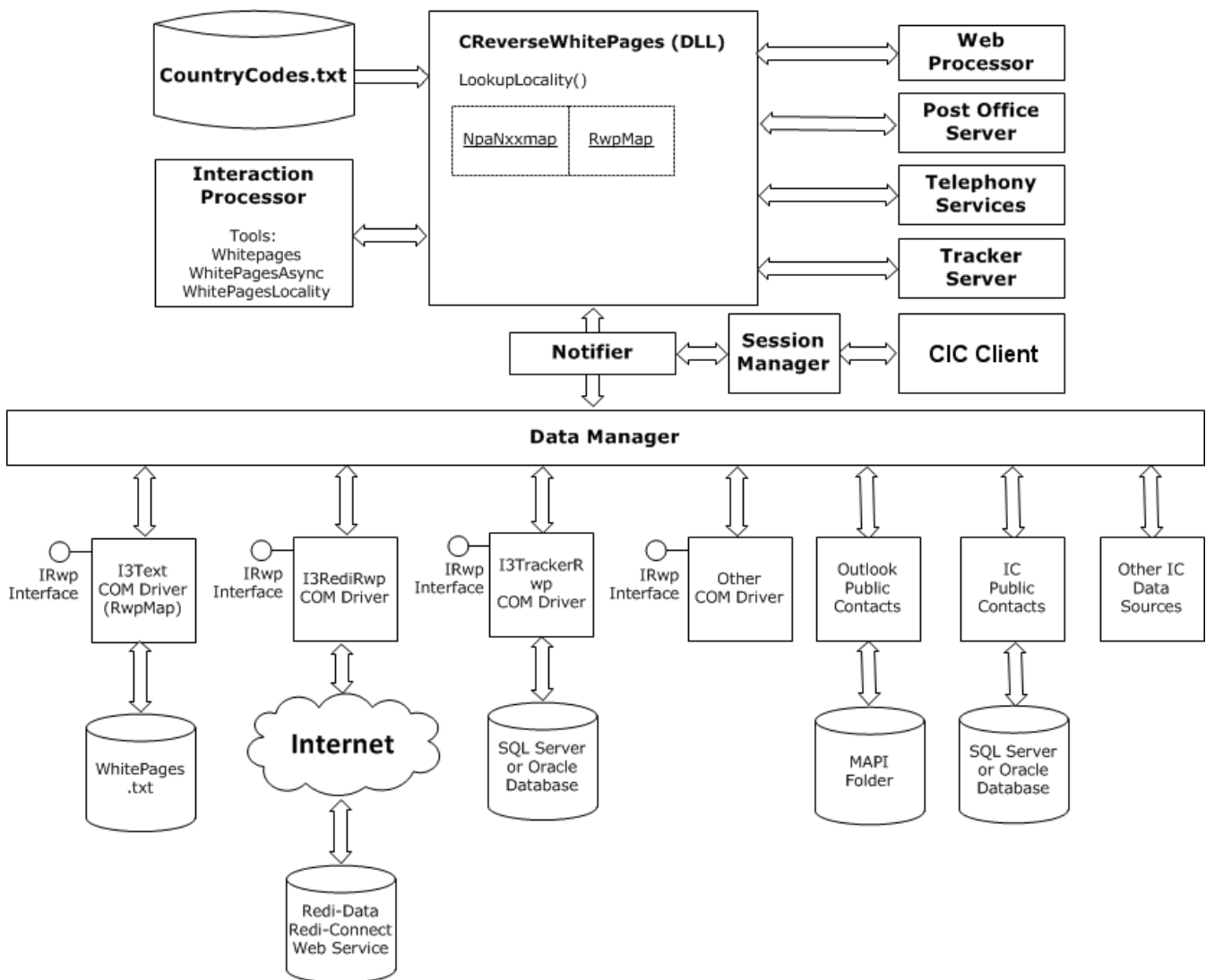
In a Customer Interaction Center environment, RWP lookups are initiated by media processing servers, such as Telephony Services (TS), by Interaction Processor (IP) handlers, and by Tracker Server.

RWP processing is divided into two main phases. Some parts of the system only initiate one phase or the other; most initiate both, where one phase is completed after the other. Those phases are:

- Locality lookup is a quick lookup against a set of internal tables inside the CReverseWhitePages DLL to get information about the caller's location (typically city and state, or city and country).
- RWP lookup is a sequenced lookup against external data sources, such as Oracle. An RWP lookup is always serviced by the Data Manager subsystem. Subsystems or applications wanting to initiate an RWP lookup invoke a CReverseWhitePages method, which uses Notifier to send/receive the request to/from Data Manager. The details of both of these phases are discussed in [RWP Call Processing Algorithms](#).

The RWP Process

A variety of [subsystems and components](#) work together whenever a CIC server performs RWP processing. The diagram below illustrates the processes and systems involved.



Subsystems and Components used when lookups are performed

Telephony Services (TS) is the CIC subsystem that works with telephony hardware to perform telephony operations, such as phone dialing. TS will initiate an RWP lookup request before placing an outbound call, and on inbound calls if ANI/DNIS information is present and was parsed successfully.

Web Processor (WP) is the CIC subsystem that handles all incoming web interactions and internal intercom chats. It operates in conjunction with a web server and acts as a web interface into the CIC system. WP issues an RWP lookup request with every new incoming web interaction.

Post Office Server (POS) is the CIC subsystem that provides platform independent access to email services such as message store access and message delivery. POS also provides support for email routing, and will initiate an RWP lookup request before queuing an incoming email interaction.

Interaction Processor (IP) is the CIC subsystem that processes low-level subsystem events in order to implement higher-level business logic. IP will initiate RWP lookups in order to obtain the contact information for use in email headers for voice mails and faxes. In addition, customers/resellers can customize IP handlers to issue RWP lookups for custom name handling, and to obtain RWP data provider-specific information.

The **CRReverseWhitePagesU.dll** component is called directly by the media processing servers (i.e. TS, WP, and POS) and by IP's RWP tool steps. This DLL performs in-memory locality lookups, and passes all RWP lookups - synchronous or asynchronous - to the DM subsystem for processing.

The **CIC client** is the user-facing component that displays the RWP lookup results (i.e. the party's name) as well as prompting for and handling manual contacts resolutions.

Data Manager (DM) is the CIC subsystem that services RWP lookup requests, as well as contact directory requests. For some RWP sources, such as Outlook Private Contacts, DM has built-in data access logic. For the other RWP sources, DM utilizes COM-based drivers to perform the actual data access. This means RWP lookups on CIC can be extended to any type of data repository for which a COM driver is available.

IRwp COM Drivers are COM objects that handle the task of communicating with an external data source, such as the white pages data service provided by Redi-Data, Inc. To be compatible with CIC, drivers developed by third parties must implement a COM interface named IRwp (see the Related Topics below).

Note: even though the Interaction Tracker database is not an external database, the RWP lookups against it are via an IRwp COM driver (I3TrackerRwp).

Related Topics

- For information about developing 3rd party IRwp drivers for the Customer Interaction Center platform, see [Developing Drivers for RWP Data Sources](#).
- To learn how to configure data sources, see [Configuring Data Sources in Interaction Administrator](#).
- For information concerning the format of text files, see [Appendix C: Text File Formats](#).

RWP Processing Algorithms

The RWP processing algorithm varies depending on whether or not Interaction Tracker is licensed. This section describes the algorithm without delving into the details of [processing performed in Interaction Tracker](#).

Note: This discussion is mainly applicable to external interactions. In general, intercom interactions do not issue RWP lookups since CIC already knows who the user is. However, there are some exceptions, such as EMS external leg calls, and handler-initiated lookups for fax and voicemail header contact information.

The following sequence of events occurs during RWP processing:

Step 1: Initial setting of EIC_RemoteName in CReverseWhitePagesU DLL

Even though the RWP lookup in DataManager typically executes in sub-second time, it is possible that, depending on the RWP source types and loading, it could take several seconds to execute. So, it is desirable to populate something in the CIC client's Name field right away, even if it is not as accurate as the final value, which will be set when the RWP lookup completes (if no match is found, then this initial Name value will be kept). The content of the Name field is governed by an interaction attribute called EIC_RemoteName.

IMPORTANT: All lookups, locality or RWP, are performed on the Standardized Number (see the IC Regionalization and Dial Plan technical reference). For example, if the number passed into the lookup was "317-872-3000" it would be standardized to "+13178723000" prior to using it for the lookups.

The first thing the RWP processing does is set EIC_RemoteName as follows:

1. If the interaction is an inbound interaction, and the carrier provides a name (e.g. Caller ID) with the incoming interaction, then that name is used to set EIC_RemoteName.

Note: currently only applicable to Telephony Services.

2. Otherwise, a locality lookup is performed. Locality lookups are divided into two phases:
 - a. Phase 1: A fast, in-memory lookup from an area code and exchange table (arranged as a hash map). This lookup returns information about the caller's location, typically city and state (e.g. "Indianapolis, IN"). The data for this table is burned into the CReverseWhitePages DLL, and cannot be modified by customers. It is typically updated with every new CIC release.

Note: currently only calls and faxes whose phone numbers are recognized as being part of the North American Numbering Plan (NANP) will trigger a locality lookup.

- b. Phase 2: If the locality lookup fails to find a match, or if the number is not part of the NANP, then a fast, in-memory lookup based on the country code and city code will be performed to get the country name and (if possible) city name. This data is stored in a file called CountryCodes.txt located in the IC resources directory, and is loaded into memory (also as a hash map) the first time CReverseWhitePages gets a lookup request for this data.

If either finds a match, then EIC_RemoteName is set to the matched display name value.

3. Otherwise, EIC_RemoteName is set to %0UNKNOWNNAME0%. (In versions before 2.4, it EIC_RemoteName was set to "Unknown Name" (localized). In 2.4 and later, RWP no longer returns "Unknown Name" when a carrier does not provide Caller ID, and when locality lookups fail to return information about the caller's location. EIC_RemoteName is set to %0UNKNOWNNAME0% instead of to "Unknown Name". This provides QMLib with a distinct string to localize. This value differs from the %nnnnn% format which might cause handlers who need to do string comparisons to hardcode a number for the string resource ("nnnnn") that could change in the future. This change guarantees that handlers are passed Unknown Name as a constant value, %0UNKNOWNNAME0%.

The endpoint (client, report, etc.) that needs to display unknown name as a localized string is responsible for localizing the value of EIC_RemoteName. This is accomplished by passing EIC_RemoteName to a function that localizes values. The QM function that replaces the escape string with a localized string is QueryLocalizedValue(). Care should be taken when calling this function on a server component that might push the string to a client. That could result in a localized value of the server and not the client.

Step 2: Asynchronous RWP lookup invocation in CReverseWhitePagesU DLL

Once EIC_RemoteName is set, the asynchronous DataManager API call to perform the RWP lookup is made. The value used to set EIC_RemoteName in #1 above is passed in as the default for Tracker (if licensed) to use in the event no match is found.

Step 3: RWP lookups in DataManagerU.exe

When DataManager receives an RWP lookup request, it performs the following steps for each RWP source configured in the RWP lookup sequence:

1. If the source is not eligible (i.e. a public source and a private lookup (or vice-versa), or a source marked for inbound lookups only and an outbound interaction (or vice-versa)) then skip to the next source in the sequence.
2. Check the source's lookup results cache. If a match is found, use the cached results.

Note: currently the Tracker RWP source and the I3Text RWP source are not cached.

3. Otherwise, perform the actual lookup (either directly or via the COM driver). If one or more matches are found, update the source's cache with the results (if not the Tracker or I3Text sources).
4. If one or more matching result rows were found (cached or non-cached), append these to any previous result rows found .
5. If this is the first source to find any matching rows, then, then make the first row the candidate result row by:
 - a. Setting EIC_RemoteName to the row's display name.
 - b. If Tracker is licensed and the RWP data source is configured for replication, then replicate this row into Tracker's database.
 - c. Sending Tracker Server (if Tracker is licensed) the candidate result row via Notifier. Tracker Server will use this information to set the Individ column of the Intx_Participant record.
6. If this source is configured as a "stop" source, or if the configured row limit is reached, then break out of this loop and proceed to step below.
7. If Tracker is licensed, steps 7 and 8 are performed. If no matching rows were found in any source, and the default display name passed in by the caller is non-empty, then synthesize a (minimal) temporary result row from the display name, and make it the candidate result (refer to a5 above), except do not set EIC_RemoteName and do not replicate. This temporary result row does not contribute to the final results referred to in 2 below.
8. Set the Eic_TrackerRWPInfo interaction attribute with the final result, which may contain zero or more result rows. This information is used by the CIC client application to decide its prompting behavior.

RWP Processing in Interaction Tracker

Interaction Tracker performs the following RWP-related tasks:

- It replicates the candidate result into Tracker's database (if replication is enabled).
- It completes the RWP processing following user input to the CIC client's contact resolution dialog, or input from Tracker Server in response to an IP handler notification.

Both tasks are performed by a Tracker DLL called TrackerTranProviderU.DLL; however, the first task runs as part of the DataManagerU.exe process; the second one runs as part of the TrackerTranServerU.exe process.

Replicating the candidate result

If the RWP data source where the candidate row was found in is configured for replication (which will cause a Replicate="1" XML attribute/value entry to be inserted in the row XML), DataManager will invoke a routine inside of the TrackerTranProviderU DLL called RwpReplication that does the following:

Note: To end-users it may appear that Tracker can resolve an interaction to an Individual, Location, or Organization. However, internally Tracker always resolves interactions to Individuals. In the event that the Individual information is not known, the RWP lookup or the user resolution resulted in an Organization or Location—a special "Unknown" Individual is created and associated with the Organization or Location. That "Unknown" Individual's IndivID is used to set the interaction participant (Intx_Participant) record's IndivID column.

1. It looks at the attributes returned with the result row XML to determine whether the matched row is for an Individual, Location, or Organization. See [Appendix A](#) for a complete list of RWP row attributes.
2. If the matched row is an Organization, it gathers the Organization information from the result row XML, and inserts a new Organization entry into Tracker if one does not already exist. It will then check to see if that Organization has a default "Unknown" Individual associated with it. If it does, then that IndivID will be used as the participant IndivID; otherwise, one will be inserted, and that (new) IndivID will be used for the participant IndivID.
3. If the matched row is a Location, it first looks to see if the result row XML references an existing Tracker Organization entry. If an Organization reference exists, it gathers the Location information from the result row XML, and inserts a new Location entry into Tracker if an existing Location with this name does not already exist. If there was not an Organization reference in the row XML, then it will insert a Tracker Organization entry if sufficient organization information exists in the row XML. Once the new Organization entry is created, it will then insert the new Location entry into Tracker (again, if this Location does not already exist). If the Organization insert fails, the location insert will not be attempted, since Tracker Locations cannot exist without a parent Organization. Finally, it will then check to see if the Location has a default "Unknown" Individual associated with it. If it does, then that IndivID will be used as the participant IndivID; otherwise, one will be inserted, and that (new) IndivID will be used for the participant IndivID.
4. If the matched row is an Individual, it looks at the result row XML to see if needs to (and can) insert a Location and/or an Organization (see step 3 above). It then gathers the Individual information from the row XML and inserts a new Individual entry into Tracker if one does not already exist.
5. For all three cases - Individual, Location, Organization - connection (iAddress) and address values will be inserted if any connection or address attributes (e.g. IndivBusinessPhone, OrgCity, etc.) are found.

Completing the RWP processing

If Tracker is licensed, one of the last things DataManager does in its RWP processing is to set an interaction attribute called Eic_TrackerRWPIInfo. The CIC client monitors for changes to this attribute for interactions in its associated user's queue. When a change occurs, it examines the attribute value which is an XML row set string to decide whether or not it needs to prompt for manual contact resolution.

It is important to know that the Tracker Server also initiates a private RWP request once the interaction gets instantiated on the user's queue. A private RWP request is basically handled just like a public RWP request by DataManager (except that it uses private RWP sources), including setting the Eic_TrackerRWPIInfo attribute.

In general, the user will be prompted if zero matches were found, or if more than one match is found. If exactly one match is found the user is not prompted; however, users can manually activate the dialog from within the CIC client if they feel the matched entry is incorrect.

If the contact resolution dialog is never presented, or if it is presented but the user ignores it long enough for it to automatically close (after the interaction de-allocates), then the candidate result becomes the final result. Nothing has to happen for this to take place; in other words, the candidate result is the final result unless otherwise superseded.

If the contact resolution dialog is presented, the user can either select an entry from the list of RWP matches (assuming it had multiple matches), run a query to get a different list of matches to select from, or create a brand new entry. In all of these cases, when the user commits to a selection or creation, the CIC client will invoke a CompleteRwp transaction request (serviced by Transaction Server), passing in the XML results string which will only contain one row: the selected or created one.

CompleteRwp transaction requests can also be generated by Tracker Server in response to a custom notification with an event string of "RWP", and a data string that is an XML result row. IP handlers (and custom applications) can use this notification to tell Tracker to identify an interaction with the supplied result information instead of any RWP lookup results. This process is described in detail in the section titled "Definitive RWP Results."

The CompleteRwp transaction does the following:

1. If the XML result row does not have an "IndivID" attribute but does have an "AppIndivID" attribute, then the value of this attribute will be used to lookup the Tracker IndivID. If "AppIndivID" is also not present, it will try "IndivExtID."
2. If the Tracker IndivID is known at this point, and the XML result originated from an application (probably an IP handler), then a check is made to see if the XML row contains the minimal amount of data to continue with the RWP processing (see the section titled "Definitive RWP Results" for the minimal required attributes). If it does not, then a database lookup will be performed to attempt to gather the missing data. This allows the application/handler logic to be as simple as possible just setting a Tracker IndivID is sufficient.
3. If the display name XML attribute, RWP_ATTR_DisplayName, is not set in the lookup results, Interaction Tracker will attempt to generate a display name using CIC's display name generation library (see [Appendix B](#)) for information on display name generation).
4. If display name is non-empty, then the interaction attribute(s) specified in the original RWP lookup call will be set with the display name value. This step will be skipped if the interaction ID is zero (i.e. a synchronous RWP lookup).
5. If the matched row was not from Tracker, and the RWP data source is configured for replication, a call to the RwpReplication routine will be made (see [Replicating the Candidate Result](#)).
6. Otherwise, a check will be made to see if the matched Tracker entry has a connection (e.g. phone number, email address, etc.) value that corresponds to the interaction address. This step will be skipped if the interaction ID is zero.
7. Tracker Server is notified of the (potentially) updated XML row string. Tracker Server will use this information to update the IndivID column of the Intx_Participant record if it is different than what was originally inserted.
8. The Eic_TrackerRWPIInfo attribute is set to the (potentially) updated XML row string. This is so that the CIC client which is still monitoring for changes to this attribute will be able to present the correct Individual detail screen, if configured to do so.

Definitive RWP Results

It is possible in some situations to customize CIC so that it can identify who the caller is without performing an RWP lookup. For example, an IVR might prompt the caller for their customer ID, or a dialing application may be reading from a customer list, etc.

In these situations, the customized portions know who the caller is, but the rest of CIC doesn't know. There are two ways of informing CIC who the caller is, one for when Tracker is licensed, and one for when it is not.

If Tracker is Licensed

If Tracker is licensed, informing CIC of the caller's identity involves sending Tracker Server a completion notification containing information about the caller. The notification parameters are:

ObjectTypeCUSTOM_NOTIFICATION

ObjectID"TrackerServer"

EventID"RWP"

DataAn XML result row. See [Developing Drivers For RWP Data SourcesLookup Method](#) for a description of the XML result format.

Note: Even though Interaction Recorder relies on Interaction Tracker data and events, it is possible for Recorder to be licensed when Tracker is not licensed. In this situation Tracker Server will still run, providing data and events as usual; however, the Tracker portions of the CIC client will be disabled.

Unlike an XML result row generated from an RWP driver, the XML row generated from an application must contain a few more attributes besides just those for determining the display name. The reason for this is that Tracker doesn't have the context available to it when being notified from an application that normally has during an RWP lookup. The required attributes are:

- InteractionAddress(the phone number, email address, etc.)
- ICInteractionID (the IC interaction (call) ID)
- At least one of the following:

IndivIDThe Tracker Individual ID.

AppIndivIDThe application ID for this individual.

IndivExtID The external ID for this individual; when supplying an IndivExtID it is recommended that an IndivExtSource also be supplied.

DisplayName e.g. "John Q. Public"

LastName Last name.

FirstName First name.

MiddleNameMiddle name. A middle initial is also acceptable.

CompanyCompany name.

The distinction between the AppIndivID and IndivExtID is subtle. The best way to describe this is with an example:A particular CRM might use GUIDs for all IDs. Customers don't really want to deal with GUIDS when, say, entering their ID in an IVR. So, in some cases you may choose to have two IDs: The actual GUID ID used by the CRM, and the logical integer ID used by the application (IndivExtID and AppIndivID, respectively). The AppIndivID is displayed in the CIC client so if you only have one application ID and want it displayed you should use AppIndivID.

Example

An example of a minimal but still very functional - XML result string is:

```
<?xml version="1.0"?>
<rows>
<row AppIndivID="1234567890" InteractionAddress="+1317222222" ICInteractionID="2101620057"/>
</rows>
```

See the description of the [Completing the Rwp transaction](#) in [RWP Processing in Interaction Tracker](#) for details about how Tracker makes use of the AppIndivID to lookup the Tracker Individual information.

Although the above XML contains everything CIC needs in order to look up the Tracker Individual, if the application actually knows the Tracker IndivID (probably not too common) it can pass that along with the information needed to generate a display name, thus avoiding another database lookup by Tracker to get this information.

Behind the scenes, Tracker Server will add IsDefinitiveResult="1" to the XML. This means that this XML result will take precedence over any XML results generated by RWP lookups, regardless of whether the RWP lookup results come in before or after the application-supplied completion notification.

Just like the completion processing that happens with a normal RWP lookup, replication can also be performed with application-initiated completion processing by setting the Replicate attribute to "1". For example:

```
<?xml version="1.0"?>
<rows>
<row Replicate="1" LastName"Public" FirstName="John" MiddleName="Q."
InteractionAddress="+1317222222" ICInteractionID="2101620057"/>
</rows>
```

This will cause a new Tracker Individual record to be inserted with the specified first, last, and middle name fields filled in. In addition, it will also cause a IndividConnection record to be inserted with the specified interaction address. Also like the normal RWP completion, it is possible to have a Location and/or an Organization record to be replicated in an application-supplied completion (see [Replicating the candidate result](#) in [RWP Processing in Interaction Tracker](#)).

Note that the check for the OrgID for a Location insertion is made after the Organization replication, so if the Organization successfully replicated, then the OrgID will be available. Connection and address values are also replicated if the corresponding attributes are found.

IMPORTANT:

All reserved XML characters (e.g. <, >, &, ' , ") must be properly escaped. For example, a < character would be escaped as:

<

Consult any standard XML documentation for the complete list of reserved characters and their escapes. For completion notification sent from IP handlers, it is highly recommended that the XML tool steps be used to construct the XML.

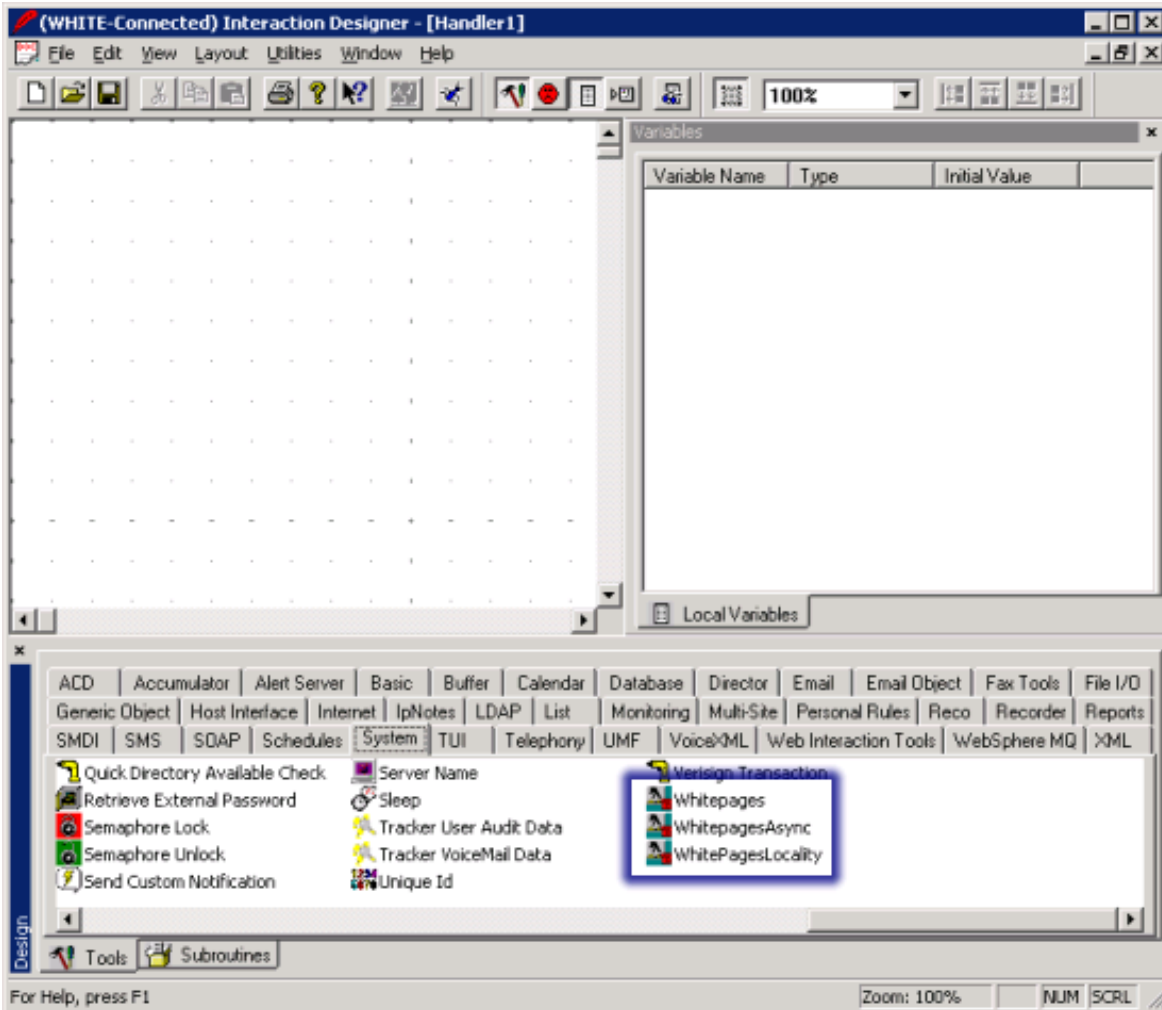
If Tracker is not Licensed

If Tracker is not licensed, informing CIC of the caller's identity is simply a matter of setting the EIC_RemoteName attribute to the caller's display name. This attribute is used to drive the display name seen in the Name field of the CIC client, as well as providing the display name for other systems (e.g. Reporting).

RWP Tools in Interaction Designer

Interaction Designer is a graphical application development tool that creates, debugs, edits, and manages handlers and subroutines, so that the behavior of the CIC server can be customized to accommodate almost any conceivable need.

Interaction Designer creates handlers that perform actions in response to an event. A handler is a collection of tool steps organized and linked to form a logical flow of actions and decisions. Each tool step is a single action that can be performed within a handler. Interaction Designer supports three tools that specifically pertain to white pages lookups. These tools appear on the System tab of the Tools palette.

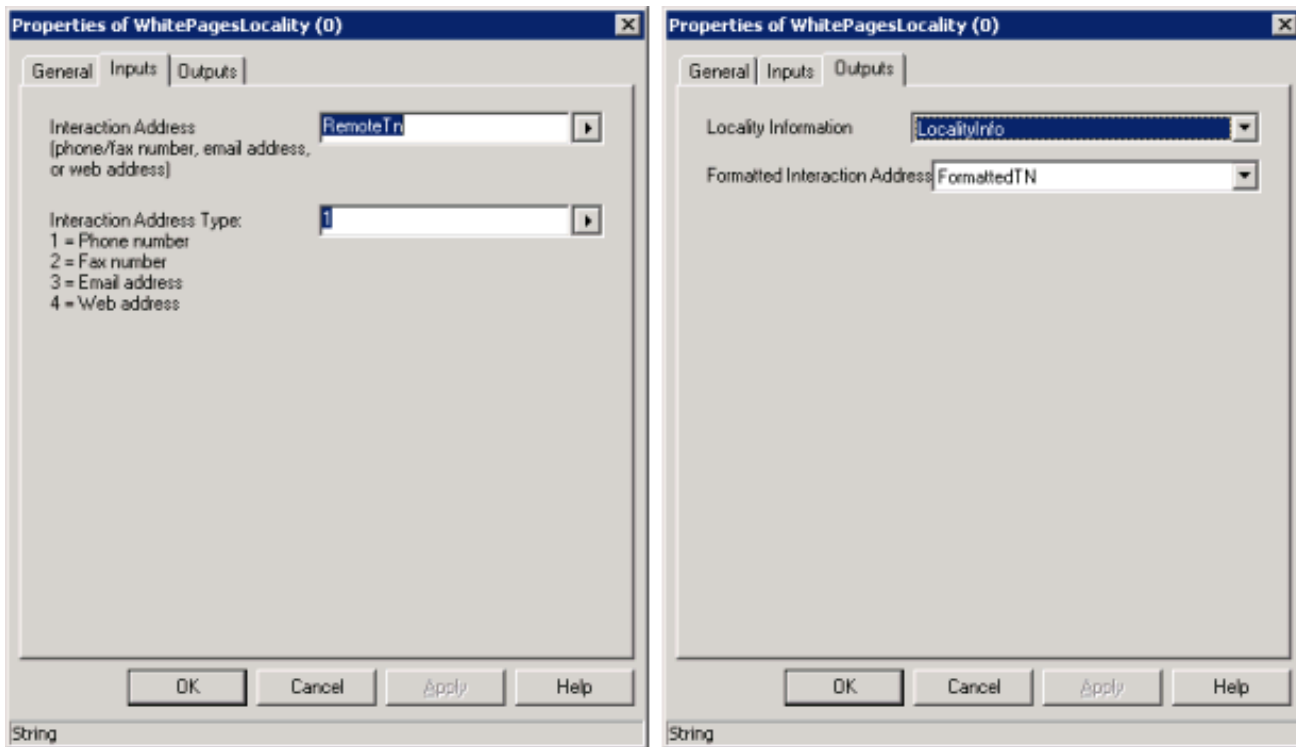


The System palette in Interaction Designer contains the Whitepages, WhitepagesAsync, and WhitePagesLocality tools. Whenever a tool is included in a handler, it is referred to as a tool step.

- The WhitePagesLocality tool returns basic locality (e.g. city & state, country, city & country, etc.).
- The WhitePages tool performs a synchronous RWP lookup, which means processing is suspended while the lookup is performed.
- The WhitePagesAsync tool performs an asynchronous RWP lookup, and assigns the result to one or more interaction attributes. Asynchronous lookups allow other processing to continue while the contact data sources are searched. When a result is returned, it is assigned to the specified attribute(s) attached to the interaction object.

WhitePagesLocality Tool

The WhitePagesLocality tool performs a very fast lookup of locality information (city/state, country, etc.) from in-memory data. See [RWP Processing Algorithms](#) for detailed information about the locality lookup process.



Inputs tab

Interaction Address

This is the interaction address (e.g. phone number, fax number, email address, web chat address, etc.) to use in the lookup.

Interaction Address Type

This specifies the type of the interaction.

Outputs tab

Locality Information

This contains the locality information (e.g. city & state, country, city & country, etc.). If no locality information could be found, it returns an empty string.

Formatted Interaction Address

For convenience, this tool also returns a formatted/displayable interaction address string.

WhitePages Tool

The WhitePages tool performs a synchronous RWP lookup, which means processing is suspended while the lookup is performed. Behind the scenes, this tool initiates a request that instructs Data Manager to perform a RWP lookup against the data sources that are configured to participate. It waits for the search to complete, and returns the results.

Properties of Whitepages (1)

General Inputs Inputs Outputs

Interaction Address: RemoteTn
(phone/fax number, email address, or web address)

Interaction Address Type: 1
1 = Phone number
2 = Fax number
3 = Email address
4 = Web address

Interaction Direction: 3
1 = Incoming
2 = Outgoing
3 = Any

Public/Private Sources: 1
1 = Public
2 = Private
3 = Both

IC User ID (ignored for Public searches):

OK Cancel Apply Help

String

Properties of Whitepages (1)

General Inputs Inputs Outputs

Contact List Sources: ""
[* separated; use "" for the default RWP sources defined in IA]

Result Format: 2
1 = Display Name
2 = Contact Summary
3 = XML

Timeout (seconds): 10.0

Send result(s) to Intraction Tracker: false

OK Cancel Apply Help

Boolean

Inputs tab

Interaction Address

This is the interaction address (e.g. phone number, fax number, email address, web chat address, etc.) to use in the lookup.

Interaction Address Type

This specifies the type of the interaction.

Interaction Direction

Specifies the direction (incoming or outgoing) of the interaction. If this lookup is not associated with an interaction, or is not direction-sensitive, specify "Any".

Public/Private Sources

Specifies the source types that will participate in the lookup. Note that if "Private" or "Both" are specified, then the IC User ID must be specified in the field below.

IC User ID

Specifies the IC User ID to use for lookups against private RWP sources.

Contact List Sources

Specifies which Data Manager Contact List sources will be used when performing this RWP lookup.

Result Format

Specifies the format of the lookup result(s) string. There are three possibilities:

Display Name

The result will simply be a single display name, as generated by the IC display name generation/formatting facility (see [Appendix B](#)). If there were multiple matched rows, then only the first row's data is used. This is the format used when setting attributes via the [WhitePagesAsync tool](#). An example display name is:

John Q. Public (XYZ Corp)

Contact Summary

The result will be a short summary of the contact data for the first matched row. This format is used for things like fax and voicemail headers, mailing addresses, etc. The first line will always be the display name, as generated by the IC display name generation/formatting facility (see [Appendix B](#)). The next lines (if present) will contain address information. For example:

John Q. Public

XYZ Corp.
Finance
1 Oak St.
Albany, NY 12345
USA

If a primary phone, email, fax, chat, or web URL is non-empty, it will also be added (and prefixed accordingly); for example, if they were all non-empty you might have:

John Q. Public
XYZ Corp.
Finance
1 Oak St.
Albany, NY 12345
USA
phone:(333) 444-5555
fax:(333) 444-5556
mailto:john.public@xyz.com
chat: john.public@xyz.com
http://www.xyx.com

XML

The result will be an XML string. This is the only format that may contain more than one result row. It is also the only format that allows applications to get access to driver-specific attributes. This is the format used internally for nearly all of the RWP processing. An example XML string that contains two result rows is:

```
<?xml version="1.0"?>
<rows>
<row IndividID="iovm~m&du#~5*|090|v3" FirstName="John" LastName="Doe"
Department="Marketing" IsPrivate="0" LocID="v9y+oec=&gaz$9f,[w^93" LocName="Acme-
Toledo" OrgID="cbrwl4h.*&#n9to$0?@&#?3" OrgName="Acme, Inc."
Phone="+12223334444^123" Email="john.doe@acme.com" InteractionAddress="+12223334444"
InteractionType="1" Source="I3Tracker Public Rwp" ExtSource="I3Tracker Public Rwp"
Replicate="0" IsStopSource="1" ICInteractionID="2100176223"
ICInteractionAttribute="Eic_RemoteName" DisplayName="Doe, John (Acme, Inc.)"
ContactSummary="Doe, John&#xA;Acme, Inc.&#xA;Marketing&#xA;phone: (222)333-4444 ext
123"&#xA; Origin="2" IsDefinitiveResult="0"/>
<row IndividID="e8x`.d#.d~;np6sxh&gt;" FirstName="Jane" LastName="Doe" Department="Sales"
IsPrivate="0" LocID="{b>8b];u(>4v{#-;388]4" LocName="Acme-Cleveland" OrgID="
cbrwl4h.*&#n9to$0?@&#?3" OrgName="Acme, Inc." Phone="+12223334444^124"
Email="jane.doe@acme.com" InteractionAddress="+12223334444" InteractionType="1"
Source="I3Tracker Public Rwp" ExtSource="I3Tracker Public Rwp" Replicate="0"
IsStopSource="1" ICInteractionID="2100176223" ICInteractionAttribute="Eic_RemoteName"
DisplayName="Doe, Jane (Acme, Inc.)" ContactSummary="Doe, Jane&#xA;Acme,
Inc.&#xA;Sales&#xA;phone: (222)333-4444 ext 124"&#xA; Origin="2" IsDefinitiveResult="0"/>
</rows>
```

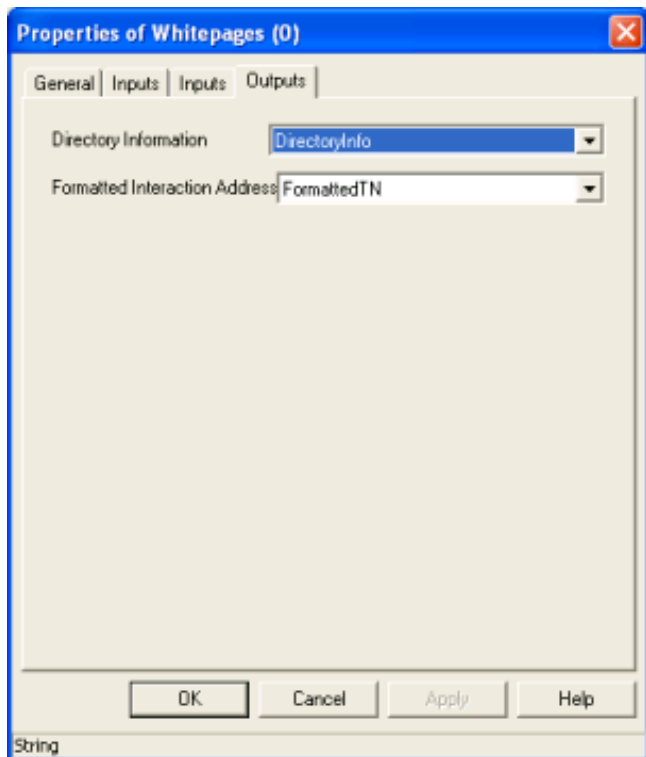
Timeout

The number of seconds for this tool to wait for a DataManager RWP lookup response.

Send Result(s) To Interaction Tracker

Whether or not the results of this lookup should be recognized by Tracker (only applicable if Tracker is licensed).

Outputs tab



Directory Information

This contains the lookup results. The format of this depends on the value of the Result Format input parameter above.

Formatted Interaction Address

For convenience, this tool also returns a formatted/displayable interaction address string.

WhitePagesAsync Tool

The WhitePages tool performs an asynchronous RWP lookup, which means processing does not wait for the lookup to be performed. Behind the scenes, this tool initiates a request that instructs Data Manager to perform a RWP lookup against the data sources that are configured to participate. The handler then continues on to execute the next step.

Inputs tab

Interaction Address

This is the interaction address (e.g. phone number, fax number, email address, web chat address, etc.) to use in the lookup.

Interaction Address Type

This specifies the type of the interaction.

Interaction Direction

Specifies the direction (incoming or outgoing) of the interaction. If this lookup is not associated with an interaction, or is not direction-sensitive, specify "Any".

Public/Private Sources

Specifies the source types that will participate in the lookup. Note that if "Private" or "Both" are specified, then the IC User ID must be specified in the field below.

IC User ID

Specifies the IC User ID to use for lookups against private RWP sources.

Call Identifier

The ID of the interaction whose specified attribute(s) will be set with the results.

Attribute Name

The name of the attribute(s) to set with the results. In general, only one attribute will be set EIC_RemoteName; however, multiple attributes can be set by listing their names, separated by '+'.

Contact List Sources

Specifies which Data Manager Contact List Sources will be used when performing this RWP lookup.

Result Format

Specifies the format of the lookup result(s) string, which will be used to set the specified interaction attribute(s). For descriptions of the supported format options, see the description of the [WhitePages tool](#) result format field.

Default Display Name

The display name to use if no match is found on this interaction address.

Send Result(s) To Interaction Tracker

Whether or not the results of this lookup should be recognized by Tracker (only applicable if Tracker is licensed).

Related Documentation

- For more information about attributes in CIC, refer to the *Interaction Attributes Technical Reference*.
- For more information about handlers, tools, and graphical program development, refer to the Interaction Designer Online Help.

Configuring Data Sources in Interaction Administrator

Data Manager makes multiple data sources appear to be a single searchable data source that CIC can access. Since RWP data requests are funneled through Data Manager, calling applications don't need to know how to access databases and other data sources directly. However, Data Manager does need to know this. When a RWP lookup is performed, Data Manager communicates with various data stores by invoking functions in custom and built-in drivers for the data source. These drivers are identified in Interaction Administrator along with the data source and its file format. Data Manager tells each driver to open, search, and close its data source. The driver returns matching data, if any, to Data Manager.

Many drivers are COM programs that handle the task of communicating with and searching a data source. Third parties who have databases they want to integrate with the Customer Interaction Center platform can develop drivers using any programming language that supports the Component Object Model. See [Developing Drivers for RWP Data Sources](#), for more details.

Install and Register COM Drivers

If you have purchased a 3rd party driver, such as Think Direct Marketing's Redi-Connect, run its setup program to install it on your CIC server. If the developer did not provide a setup program, register the COM component manually by entering regsvr32 filename.ext in a command window.

Configure Data Sources for new Drivers

The procedures in this section explain how to configure RWP data sources in Interaction Administrator. To configure a RWP Data source, you need to complete three high-level steps. These steps are explained in detail .

Start Interaction Administrator. Then [Configure an IC Data Source](#) .

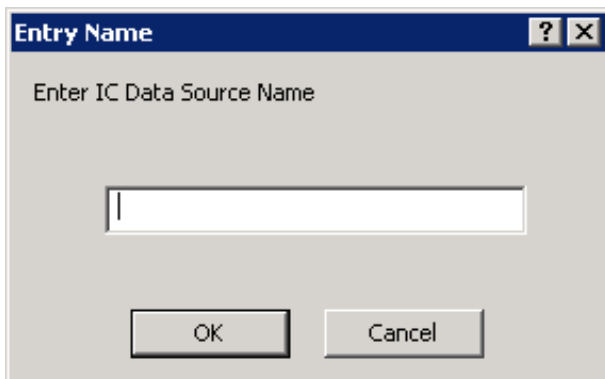
Use Interaction Administrator to [Configure a Data Manager Contact List Source](#).

[Add the source to the RWP Lookup Sequence](#).

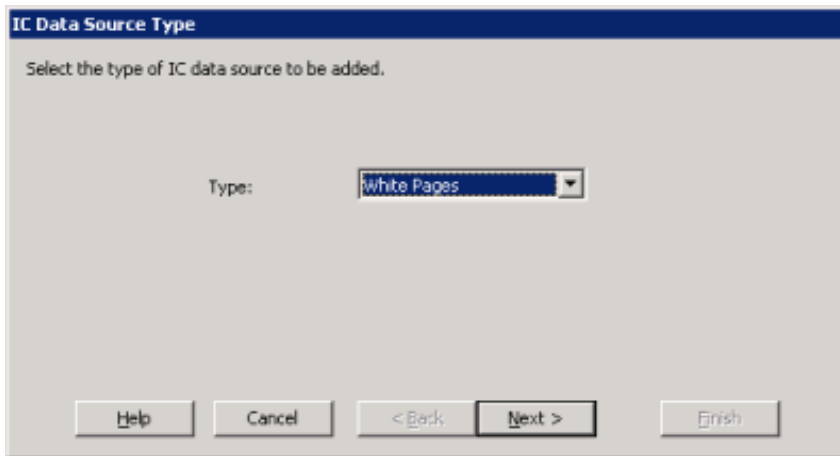
Configure an IC Data Source

To create and configure the IC data source, perform the following steps:

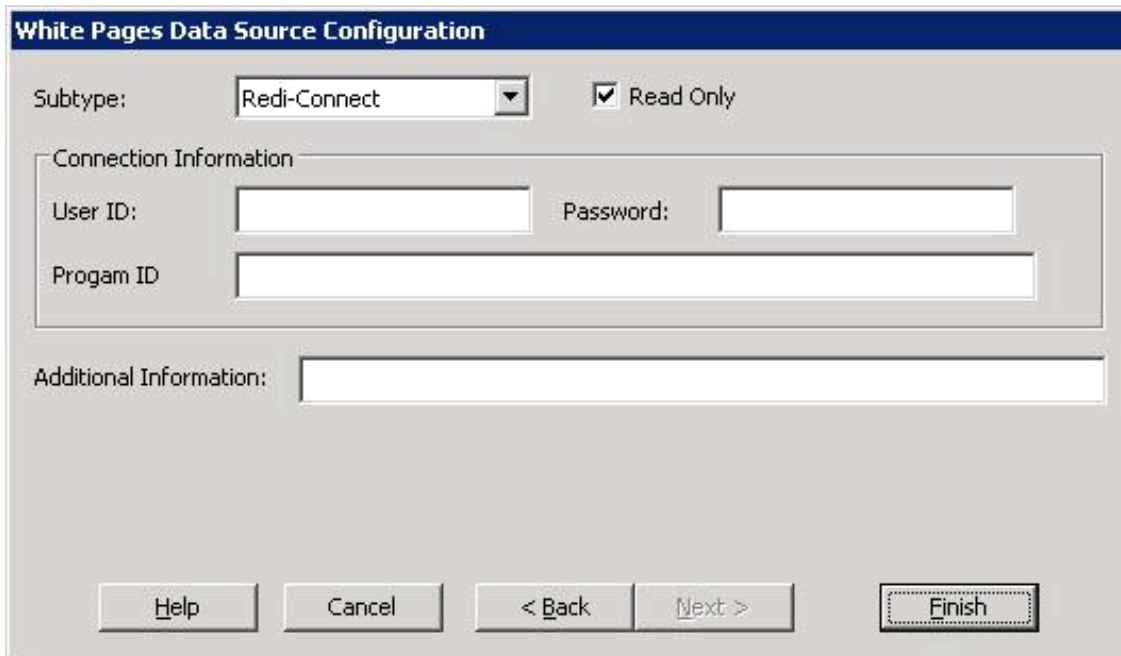
1. In Interaction Administrator, highlight the IC Data Sources container.
2. Press the Insert key to add a new IC Data Source.



3. Enter a descriptive name for the data source.
4. Click OK.
5. The IC Data Source Type dialog appears as shown below. Select White Pages for the type.



6. Click Next.
7. The White Pages Data Source Configuration page appears. Use the Subtype list to identify the type of driver you want to add.



If you are adding the driver provided by Redi-Data, Inc.:

- a. Select Redi-Connect from the Subtype drop list
- b. Type your Redi-Connect ClientID in the User ID field. This information will have been supplied to you by Redi-Data.
- c. No additional information needs to be specified to configure this driver. Leave the other fields blank.

If you are adding a driver for the WhitePages.txt file:

- a. Select I3 Text File from the Subtype drop list.
- b. Leave the User ID, Password, and Program ID fields blank.
- c. The Additional Information field may be left blank. However, if the name of your whitepages file is something other than Whitepages.txt, and/or it is not located in the <resourcePath> directory (where <resourcePath> is the path configured in Interaction Administrator), you can add a WhitePagesFile entry that identifies the explicit path to the file. For example, you might enter the following entry in the Additional Information field:

WhitePagesFile=D:\MyData\MyWhitePages.txt;

If you are adding an unlisted third-party driver, select Other.

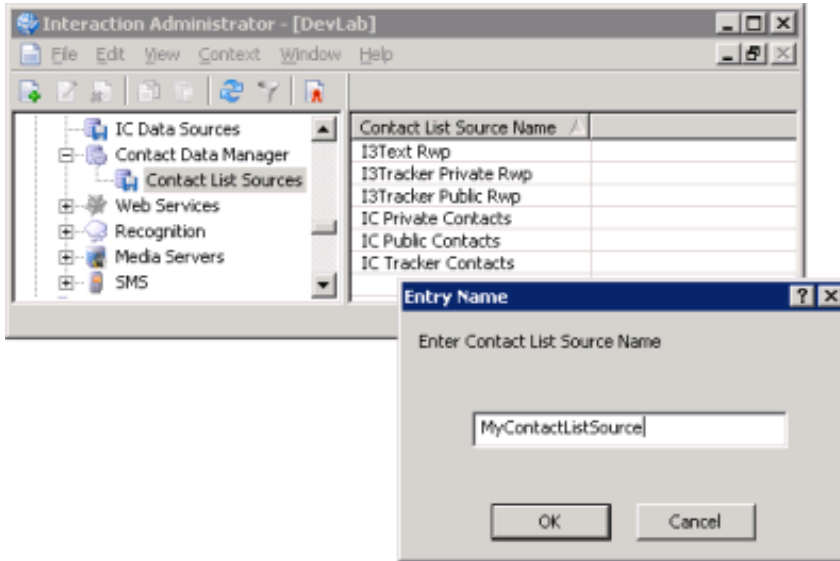
- a. If the third-party driver requires a User ID and Password to be sent to it, enter this information into the respective fields.
- b. Fill in the Program ID field. This is required for all drivers of the "Other" subtype. To do this, type the COM Program ID that corresponds to this driver or the Class ID (CLSID) of the component, inside curly braces. A typical Class ID entry might be: {EE7B00C3-6974-485C-A7A9-3D62E8F9C614}. If you do not know the Program ID or Class ID, contact the vendor of the 3rd-party driver.
- c. Optionally use the Additional Information field to supply any other information that your driver may require using ATTRIBUTE=VALUE; syntax.

8. If this data source is read-only (as most white page sources are), make sure the Read Only checkbox is checked.
9. Click OK to close the IC Data Source Configuration dialog.

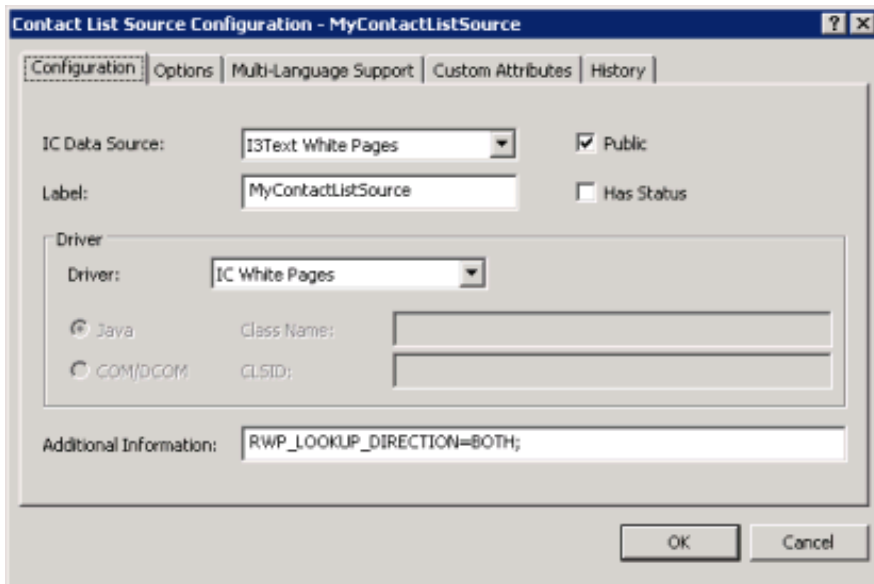
Configure a DataManager Contact List Source

To create and configure a Data Manager Contact List source, perform the following steps:

1. In Interaction Administrator, highlight Contact List Sources under the Contact Data Manager container.
2. Press the Insert key to add a new source. This opens the Entry Name dialog, as shown below:



3. Enter a descriptive name for the new Contact List Source.
4. Click OK.
5. For the IC Data Source, select the IC data source name you entered back in Step 1, item



6. Make sure that the Public checkbox is checked.
7. For the driver, select IC White Pages.
8. Your driver may require specific information to be entered in the Additional Information field. The data items that you enter here are driver-specific ATTRIBUTE=VALUE; pairs that your driver requires on a per list-source basis. You can also enter ATTRIBUTE=VALUE; pairs for IC-specific options. Currently, only one IC-specific option is supported:

RWP_DIRECTION

This option specifies whether lookups will be performed for incoming and outgoing calls. Valid values for this attribute are "INBOUND", "OUTBOUND", OR "BOTH".

INBOUND indicates that RWP lookup should only be performed on inbound calls.

OUTBOUND indicates that the RWP lookup should only be performed on outbound calls.

BOTH is the default. The RWP lookup is performed on both inbound and outbound calls.

For example, valid entries in the Additional Information field might be:

```
RWP_DIRECTION=INBOUND;
```

```
RWP_DIRECTION=OUTBOUND;
```

```
RWP_DIRECTION=BOTH;
```

RWP_ADDRESS_TYPE

This option specifies which interaction address types will be searched. The default is to search all types, which are:

AssistantPhone

BusinessEmail

BusinessPhone

BusinessPhone2

Fax

HomeEmail

HomePhone

HomePhone2

Mobile

Pager

If the number of contacts in your source is large (e.g.: over two thousand), you might consider limiting the number of types to search on and/or add indexes to the phone number fields you are searching on.

Note: the Tracker RWP source does not use this option, since it searches the IndividConnection table (and possibly LocConnection and OrgConnection tables), which store connection (address) values of all types.

To specify exactly which interaction types to search on, simply list them with a comma separator. For example:

```
RWP_ADDRESS_TYPES=BusinessEmail,HomePhone2;
```

RWP_TIMEOUT

RWP_TIMEOUT is the amount of time, in seconds, that DataManager will wait on the driver to complete its lookup. Note for RWP purposes, a timeout is not considered an error (it is treated as a no-match situation). The default timeout is 10 seconds. For example, to set the timeout to 30 seconds you would have RWP_TIMEOUT=30;

RWP_ROWLIMIT

This is the limit on the number of result rows for this data source. The default limit for Tracker RWP sources is 100 rows; for all other sources it is 1 row. For example, to set this source's row limit to 50, you would have the following:

```
RWP_ROWLIMIT=50;
```

RWP_REPLICATE_MATCHES

This determines whether or not to replicate matches found in this data source into Tracker's database. The default is true. For example, to disable the automatic replication, you would have the following:

```
RWP_REPLICATE_MATCHES=False;
```

RWP_STOP_ON_MATCHES

This determines whether or not to break out of the RWP search sequence if one or more matches are found in this data source. The default is true. For example, to have the searching continue on after matching, you would have the following:

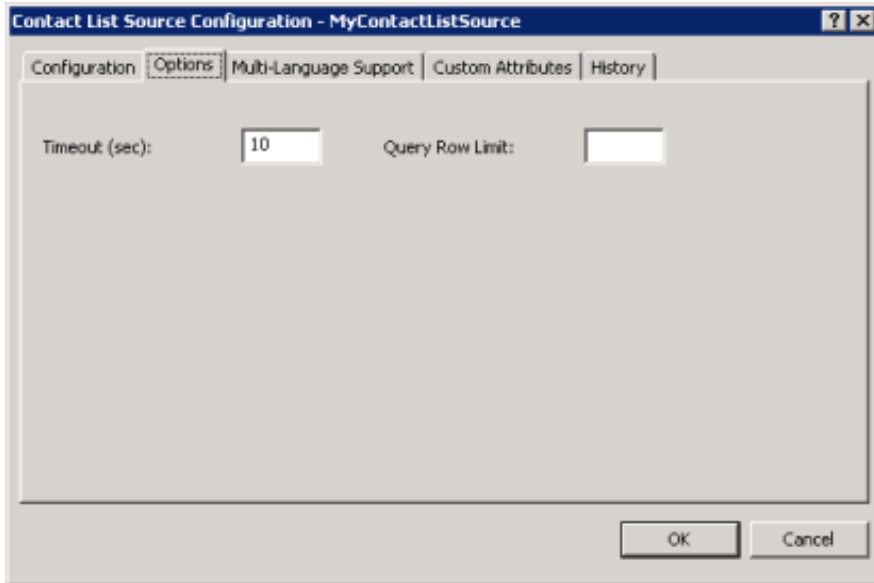
```
RWP_STOP_ON_MATCHES=False;
```

RWP_DRIVER_CLSID

This contains the CLSID or ProgID of the RWP driver. Note that this is for RWP drivers that are not intrinsically tied to an IC Data Source, like the I3 TDM RWP driver is. Currently, only the Tracker RWP data sources use this option. Since this is pre-configured, you generally will not have to manually configure this option.

None of the I3 RWP drivers have driver-specific configuration options that must be configured in this field. It is unlikely that custom drivers will have custom options, since it is anticipated that they will use the IC Data Source's Additional Information field instead. So, most of the time this field will be left blank for RWP-only list sources.

9. Click on the Options tab.

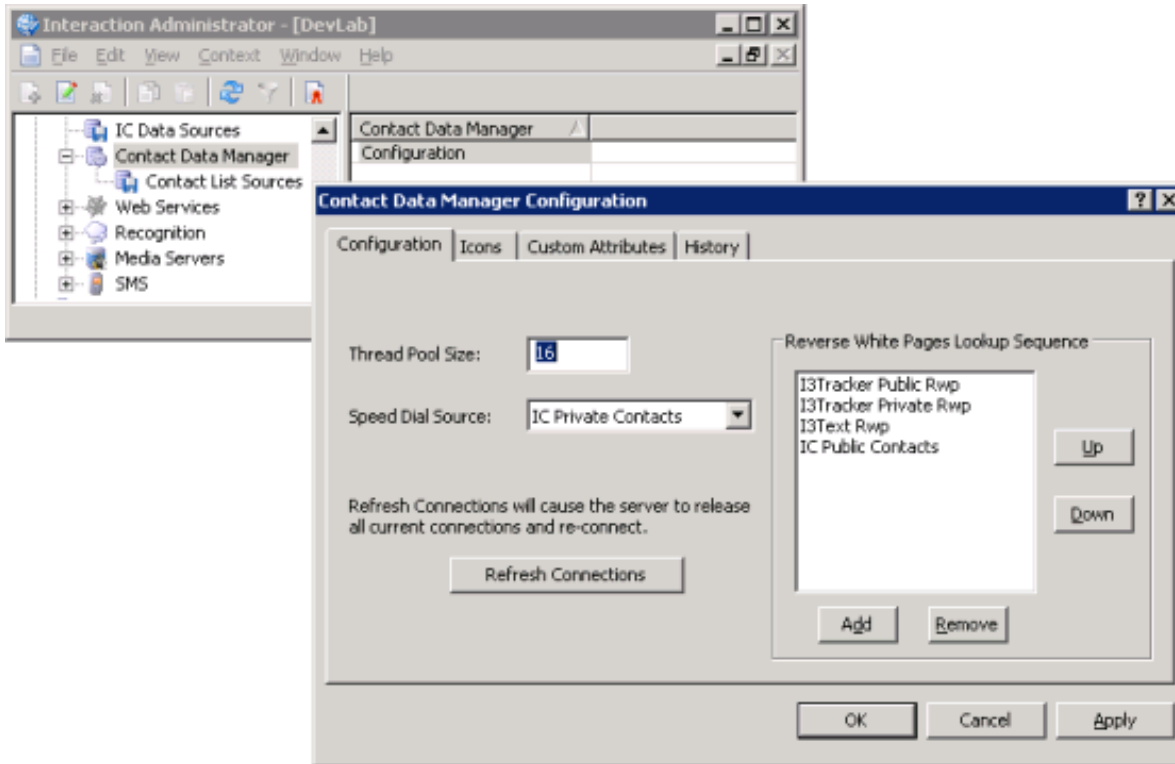


10. Enter a timeout value in the Timeout (sec) field. A value of 10 seconds is appropriate in most cases. This timeout is for contact directory loading in the CIC client; it is not the timeout used for RWP processing (see the RWP_TIMEOUT attribute above).
11. You can leave the Query Row Limit field empty. This row limit is for contact directory loading in the CIC client; it is not the row limit used for RWP processing (see the RWP_ROWLIMIT Additional Information field attribute documented above).
12. Press OK to close the dialog.

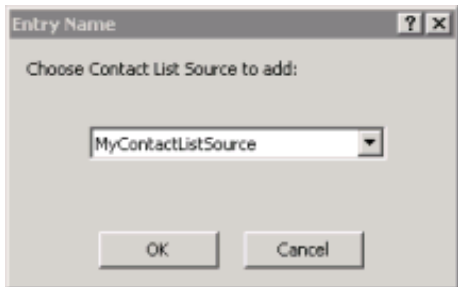
Add the source to the RWP Lookup Sequence

To add your new contact list source to the lookup sequence, perform the following steps in Interaction Administrator:

1. Highlight the Contact Data Manager node.
2. Double-click Configuration in the right-hand pane to open the Contact Data Manager Configuration dialog.



3. In the Reverse White Pages Lookup Sequence frame, click the Add button. This opens the Entry Name dialog shown below.



4. Select the name of the contact list source you created in Step 2 above. Then press the OK button. Your new Contact List Source will appear in the list of sources in the Reverse White Pages Lookup Sequence frame.
5. Select a list item, and use the Up/Down buttons, reorder items in the list. As a rule of thumb, you should place your fastest sources at the top. You should also consider not including some very slow sources, such as Outlook contacts. We recommend that you place I3Text Rwp source at the very top, so that it can act as an override source.
6. Press the OK button to save your changes.

Developing Drivers for RWP Data Sources

This section is for developers who need to develop drivers that implement the IRwp COM interface. Besides the built-in Data Manager drivers (e.g. IC Public Contacts), CIC allows externally created drivers to participate in RWP lookups. CIC ships with two default external RWP drivers: one for the WhitePages.txt file, and one for accessing Think Direct Marketing's white pages data. In addition, if Tracker is licensed, a driver for Tracker's database will also be provided. These drivers must be COM drivers; the interface they must implement is the IRwp interface. Examples below are for C and C# respectively.

IDL file for the IRwp interface - C Interface Definition

```
// Rwp.idl : IDL source for rwp lookup interface
//
// This file will be processed by the MIDL tool to
// produce the type library (Rwp.tlb) and marshalling code.
import "oaidl.idl";
import "ocidl.idl";
[
object,
uuid(703B84B8-02AB-41E3-BF61-3E63C7946CD4),
dual,
helpstring("IRwp Interface"),
pointer_default(unique)
]
interface IRwp : IDispatch
{
[id(1), helpstring("method Initialize")]
HRESULT Initialize([in] BSTR strInitializationData);
[id(2), helpstring("method Lookup")]
HRESULT Lookup([in] BSTR bstrPhoneNumber, [out, retval] BSTR * pbstrName);
[id(3), helpstring("method Deinitialize")]
HRESULT Deinitialize();
};
```

IDL file for the IRwp interface - C# Interface Definition

```
// Rwp.idl : IDL source for rwp lookup interface
[Guid("703B84B8-02AB-41E3-BF61-3E63C7946CD4")]
public interface IRwp
{
[DispId(1)]
void Initialize(string strInitializationData);
[DispId(2)]
string Lookup(string bstrPhoneNumber);
[DispId(3)]
void Deinitialize();
}
```

Data Manager uses a pool of Single-Threaded Apartment (STA) threads. Each thread creates its own instance of the RWP driver. The three functions that need to be implemented by the driver are: Initialize, Lookup, and Deinitialize.

Initialize Method

This function is called only once, immediately after the object is created. It has one input parameter, which is a BSTR containing configuration data from the IC Data Source and the Data Manager Contact List Source. This data is formatted into <ATTRIBUTE>=<VALUE> pairs, with each pair separated by a semicolon. The first two attributes are always UID and PWD, which are the User ID and Password from the IC Data Source.

Following this are any <ATTRIBUTE>=<VALUE> pairs in the Additional Information field of the IC Data Source, followed by any <ATTRIBUTE>=<VALUE> pairs in the Additional Information field of the Data Manager Contact List Source. It is in these Additional Information fields where third-party drivers can add their own configuration options. An example configuration string could be:

```
"UID=fred;PWD=;host=MyHost"
```

In this case, the host attribute is a driver-specific attribute. If the driver does not need any special initialization, it should simply do nothing and return S_OK.

Lookup Method

This function takes a phone number BSTR as its input, and sets the results of the lookup query in its output BSTR. The input number will have been converted to the CIC standardized format. For example:

```
+13178723000
```

Note that standardized numbers may have additional dialing information and/or extension information following the base number. For example:

```
+13178723000/^131
```

So, unless the driver is performing a sub string match, it should truncate after the 12th character for NANP numbers (most likely drivers will also need to strip the leading "+1").

The output BSTR should be set to empty on failure to find a match. In this case the HRESULT return value should still be S_OK, since a failed match is not considered an error. On a success lookup the driver should set the output BSTR as outlined below:

It should be formatted as an XML document string. The basic structure is:

```
<?xml version="1.0"?>
<rows>
</rows>
```

In fact, this is exactly what a driver should return in the case of no matching rows.

Note: returning an empty string will also have the same effect (i.e. DataManager will convert an empty string to the above XML). For each row returned these will be a <row> element.Attributes are used to hold the lookup result values. So, for example, a result with two matching rows might look like:

```
<?xml version="1.0"?>
<rows>
<row IndividID="iovg*w~m&du#~5*|090|v3" FirstName="John" LastName="Doe" Department="Marketing"
IsPrivate="0" LocID="v9y+oec=&gaz$V9f,[w^93" LocName="Acme-Toledo"
OrgID="cbrwl4h.*&#n9to$0?@&#?3" OrgName="Acme, Inc." Phone="+12223334444^123"
Email="john.doe@acme.com" InteractionAddress="+12223334444" InteractionType="1"
Source="I3Tracker Public Rwp" ExtSource="I3Tracker Public Rwp" Replicate="0" IsStopSource="1"
ICInteractionID="2100176223" ICInteractionAttribute="Eic_RemoteName" DisplayName="Doe, John
(Acme, Inc.)" ContactSummary="Doe, John&#xA;Acme, Inc.&#xA;Marketing&#xA;phone: (222) 333-4444
ext 123"&#xA; Origin="2" IsDefinitiveResult="0"/>
<row IndividID="e8x`.d#.*d~;np6sxh&gt;" FirstName="Jane" LastName="Doe" Department="Sales"
IsPrivate="0" LocID="{b>8b];u(>4v{#-;388]4" LocName="Acme-Cleveland" OrgID="
cbrwl4h.*&#n9to$0?@&#?3" OrgName="Acme, Inc." Phone="+12223334444^124"
Email="jane.doe@acme.com" InteractionAddress="+12223334444" InteractionType="1"
Source="I3Tracker Public Rwp" ExtSource="I3Tracker Public Rwp" Replicate="0" IsStopSource="1"
ICInteractionID="2100176223" ICInteractionAttribute="Eic_RemoteName" DisplayName="Doe, Jane
(Acme, Inc.)" ContactSummary="Doe, Jane&#xA;Acme, Inc.&#xA;Sales&#xA;phone: (222) 333-4444 ext
124"&#xA; Origin="2" IsDefinitiveResult="0"/>
</rows>
```

IMPORTANT: All reserved XML characters (e.g. <, >, &, ', ") must be properly escaped. For example, a < character would be escaped as:

<

Please consult any standard XML documentation for the complete list of reserved characters and their escapes.

There are no required attributes. However, drivers should return enough information such that DataManager can construct a display name using CIC's display name formatting facilities (see [Appendix B](#)). Basically, this means returning one or more of the following attributes:

- FirstName
- MiddleName
- LastName
- Company

Alternatively, drivers can set the DisplayName attribute directly; in which case, it will be preserved. However, this is not recommended unless the driver-originated display name potentially contains different/better information than the display name that is generated via CIC's display name formatting facilities.

Again, this is the minimal amount of information. Ideally, drivers would return as much information as possible. See [Appendix A](#) for the complete list of attributes.

Deinitialize Method

This function is called only once, immediately prior to the object being destroyed (which is usually immediately prior to the thread terminating). It takes no parameters. If the driver does not need any special de-initialization, it should simply do nothing and return S_OK.

Appendix A: RWP XML Attributes

Attributes defined by PureConnect

This topic lists all RWP XML attributes that are defined by PureConnect.

Note:

Drivers may add their own (driver-specific) attributes—see driver-defined attributes.

Control and interaction-specific attributes (3rd-pary drivers typically do not set these):

ContactSummaryDisplayName
ICInteractionAttributeICInteractionID
InteractionAddressInteractionDirection
InteractionTypesDefinitiveResult
IsPrivateIsStopSource
OriginOwner
ReplicateSource

Individual attributes:

AssistantName AppIndivID (only applicable if Tracker is licensed)
CompanyDepartment
FirstNameGender
ICUserIDIndivExtID
IndivExtSource IndivID(only applicable if Tracker is licensed)
IndivImage
IndivRemarks IndivTitleID (only applicable if Tracker is licensed)
IndivURLIndivTypeID (only applicable if Tracker is licensed)
JobTitle LastName
MiddleNameStationName
WebLoginWebPassword

Location attributes (only applicable if Tracker is licensed):

AppLocIDLocExtID
LocExtSourceLocID
LocNameLocRemarks

Organization attributes (only applicable if Tracker is licensed):

AppOrgIDOrgExtID
OrgExtSourceOrgID
OrgNameOrgRemarks
OrgTypeID

Common Individual, Location, and Organization attributes:

Depending on the context (values of other attributes present), these attributes will be targeted for an Individual, or (if Tracker is enabled) a Location or Organization. 3rd-party drivers typically do not set these attributes.

ExtID

ExtSource

Remarks

Individual address and connection attributes:

IndivBusinessStreetAddressIndivBusinessCity

IndivBusinessStateIndivBusinessZip

IndivBusinessCountryIndivHomeStreetAddress

IndivHomeCityIndivHomeState

IndivHomeZipIndivHomeCountry

IndivBusinessEmailIndivBusinessPhone

IndivBusinessPhone2IndivBusinessMobile

IndivBusinessFaxIndivBusinessPager

IndivBusinessChatIndivBusinessUrl

IndivAssistantPhoneIndivHomePhone

IndivHomeEmailIndivHomePhone2

IndivHomeMobileIndivHomeFax

IndivHomePagerIndivHomeChat

IndivHomeUrl

Location address and connection attributes (only applicable if Tracker is licensed):

LocBusinessChatLocBusinessCity

LocBusinessCountryLocBusinessEmail

LocBusinessFaxLocBusinessMobile

LocBusinessPagerLocBusinessPhone

LocBusinessPhone2LocBusinessState

LocBusinessStreetAddressLocBusinessUrl

LocBusinessZip

Organization address and connection attributes (only applicable if Tracker is licensed):

OrgBusinessChatOrgBusinessCity

OrgBusinessCountryOrgBusinessEmail

OrgBusinessFaxOrgBusinessMobile

OrgBusinessPagerOrgBusinessPhone

OrgBusinessPhone2OrgBusinessState

OrgBusinessStreetAddressOrgBusinessUrl

OrgBusinessZip

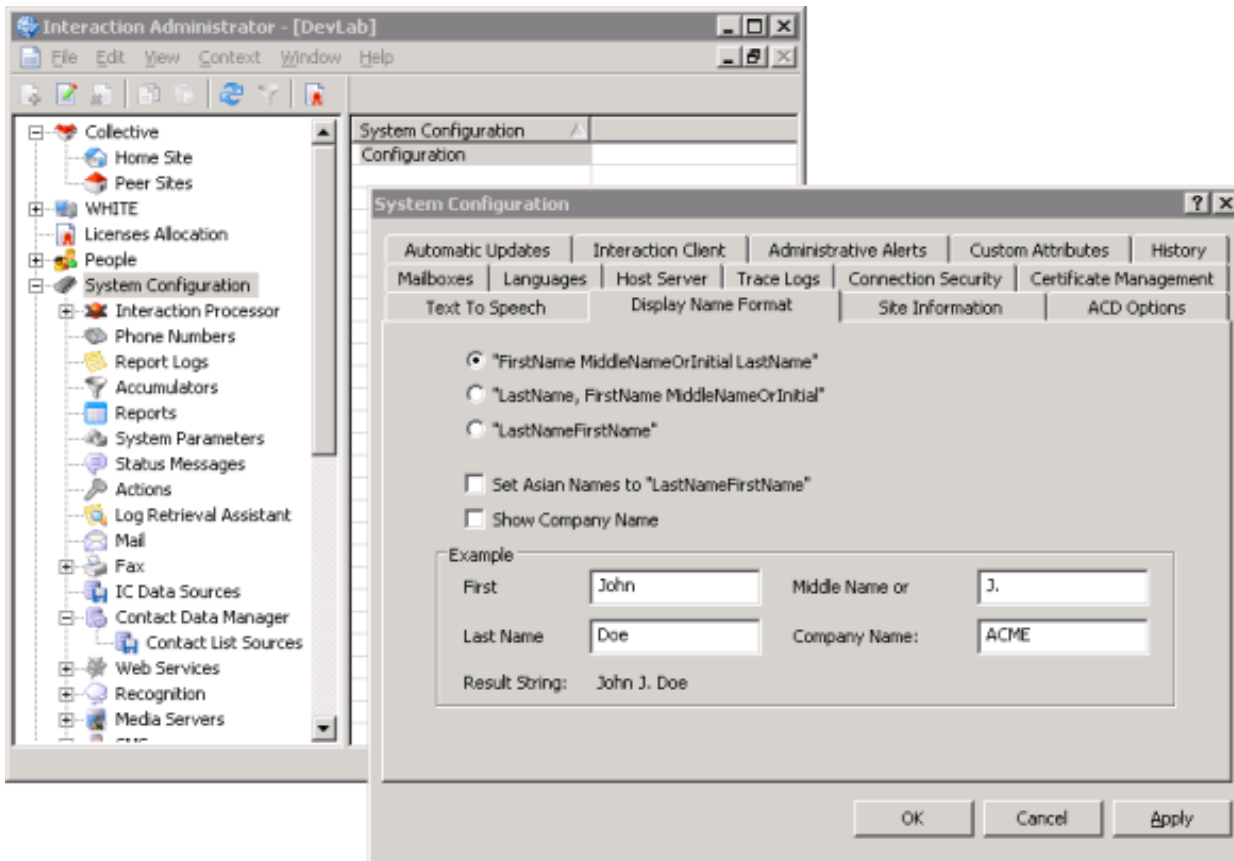
Driver-Defined Attributes

Drivers may add additional, driver-specific attributes to the XML result rows. It is up to handlers or custom applications to take advantage of any such additional attributes. To simplify the task of querying attributes in a handler or custom application, custom RWP drivers should supply I3-defined attributes in addition to driver-specific attributes, even though this may create some data redundancy.

Note: if a custom driver returns a `DisplayName` attribute, it will be used as the display name. Otherwise, the display name will be generated based on the values of other attributes, which is preferred. See [Appendix B](#) for more information on CIC's display name generation facilities.

Appendix B: Display Name Generation

Use the Display Name Format tab in Interaction Administrator to configure the way names are displayed in the system. This property sheet appears when you select the System Configuration container, and then double-click the Configuration entry.



Selecting RWP display name format in Interaction Administrator.

The radio buttons on this property sheet select the format used to display first, middle, and last names. You can type sample names in the test fields at the bottom of the page to see view the result of your selections.

"FirstName MiddleNameOrInitial Last Name" radio button

This option displays John J. Doe.

"LastName, FirstName, MiddleNameOrInitial" radio button

This option displays Doe, John J.

"LastNameFirstName" radio button

This option displays DoeJohn.

Set Asian Names to "LastNameFirstName" checkbox

Use this option to set LastNameFirstName as the default format if an Oriental name is detected.

Show Company Name checkbox

Select this option to include the Company Name in the display, if a company name is found during the lookup.

Appendix C: Text File Formats

This appendix describes text files used in conjunction with RWP lookups.

CountryCodes.txt file

CIC uses country codes to perform quick locality lookups. Country codes reside in an ASCII text file named CountryCodes.txt. A country code is a 1 to 3-digit number that precedes the national number in an international telephone call. In many cases, a city code follows the country code. To dial a number outside of the United States, the format is:

```
011 country code city code national number
```

To place a call to Benhas, Egypt, for example, one might dial:

```
011 20 13 national number
```

...where 20 is the country code for Egypt, and 13 is the number assigned to the city of Benhas. The 011 signifies that an internal call is being placed from the U.S. If the country code is 808 or 809, the dialing prefix is 1, rather than 011.

The CountryCodes.txt file resides in the <resource path> directory. The location of the <resource path> directory is specified in Interaction Administrator. It is normally set to \3\IC\Resources. Since CountryCodes.txt is an external file, it can be localized as necessary.

The information in CountryCodes.txt may be searched when a locality request is performed. See [RWP Processing Algorithms](#), for detailed information about the locality lookup process.

CountryCodes.txt File Format

Each line of data in countrycodes.txt contains a plus-character preceded country code, which is optionally followed by a city code, one or more spaces or tabs, and then name of the city and country.

Note: although not recommended, you can omit the leading '+' as long as every entry omits it.

The first few lines of the default CountryCodes.txt file are listed below, along with the comments that appear at the top of the file. This is not a complete listing, but serves to show the format of the file.

```
#Customizing CountryCodes.txt
#
# Customize this file if you want CIC to display country information associated
# with a telephone number. This file is used by the WhitePagesLocality tool
# when a more specific locality information (i.e. City, State) is not found.
#
# You can update CountryCodes.txt file while the CIC system is running; changes
# to it will take effect shortly after the file is updated. The file consists
# of one entry per line; each entry consists of a partial (or full) telephone
# number followed by whitespace followed by the country information.
#
# All lookups are done on a longest-matching leading string basis. So for
# example, a number beginning with +6724 matches Christmas Island but a number
# beginning with +6725 (which has no entry in the table) matches Antarctica,
# since Antarctica has an entry of +672 and is thus the longest match available.
#
+20Egypt
+2013Benhas, Egypt
+202Cairo, Egypt
+203Alexandria, Egypt
```

```
+2040Tanta, Egypt
+2045Damanhour, Egypt
+2048Shebin El Kom, Egypt
+2050El Mansoura, Egypt
+2062Suez, Egypt
+2066Port Said, Egypt
+2088Asyut, Egypt
+2093Sohag, Egypt
+2095Luxor, Egypt
+2096Oena, Egypt
+2097Aswan, Egypt
+210Morocco
+212Morocco
+2122Casablanca, Morocco
+21244Marrakech, Morocco
+2127Rabat, Morocco
+21299Tangiers, Morocco
+213Algeria
+216Tunisia
+2161Tunis, Tunisia
+2167Kairouan, Tunisia
+218Libya
+21821Tripoli, Libya
+21823Zawia, Libya
+21851Misurata, Libya
```

WhitePages.txt file

The information in WhitePages.txt will be searched (after being cached into memory) when an RWP lookup is performed against the I3Text White Pages data source, which uses the I3TextRwpU.dll COM diver. See [RWP Processing Algorithms](#) for detailed information about the RWP lookup process.

As with previous versions of CIC, an ASCII text file named whitepages.txt is placed in the <resource path> directory.

Whitepages.txt File Format

The format of the WhitePages.txt file is identical to previous versions, with the exception that the plus character now precedes country codes by default. Each line contains a phone number, followed by one or more spaces or tabs, followed by the name of the entry, optionally followed by a pipe ('|') character followed by address information, which can contain embedded pipe characters. CIC converts the pipe characters to new lines. The syntax is:

```
+telephone number Entry Name|Street Address|City, State Zip
```

The best way to understand this is to examine data in the sample WhitePages.txt file.

Note: In order to make the entries more readable, the entries have been split across two or three lines in the sample listing below. In the "real" WhitePages.txt file all entries are on one line.

```
#Customizing WhitePages.txt
#
# Customize this file if you want CIC to display directory information
```

```
# associated with a telephone number. The first line of directory information
# is displayed with a call in Interaction Client and the entire directory
# listing is displayed in the body of a fax or voice mail message.
#
# The information in WhitePages.txt file is divided into two segments. The left
# segment contains telephone numbers or area codes, and the right segment
# contains directory information.
#
# CIC uses this directory information in two ways. First, when a telephone call
# is displayed in a queue, CIC looks up the number and displays the first line
# of directory information (up to the first | symbol) with the call in
# Interaction Client (via the Eic_RemoteName call attribute).
#Note: This is the way it was done in 1.x, and it is still supported;
#however, we recommend that you modify the data to include the
#appropriate attribute names (see the paragraph immediately below).
# Second, CIC displays all of the directory information in the body of a fax or
# voice mail message; CIC retrieves this information with the WhitePagesAsync
# tool in a number of different handlers.
#
# It is possible for this information to be used outside of CIC. For example, to
# populate a set of fields for a client-side screen pop. In this case the
# original (CIC 1.x) format of the data doesn't provide the needed attribute
# information. So for example, instead of:
#+1-844-274-5992Genesys Telecommunications Lab, Inc.|2001 Junipero Serra Blvd|Daly City, CA
94014|USA
# you would have:
#+1-844-274-5992DisplayName: Genesys Telecommunications Lab, Inc.|Company: Genesys|Address: 2001
Junipero Serra Blvd|City: Daly City|State: CA|Zip: 94014|Country: USA
# Keep this in mind when migrating any 1.x WhitePages.txt data to a newer
# release.
#
# The WhitePages tool in Interaction Designer takes a Telephone Number as
# input, and returns the directory information as a string. If the directory
# information contains multiple lines (indicated by pipe (|)characters), each
# pipe is treated as a newline in the fax and voice mail forms. See the
# WhitePages tool documentation in Interaction Designer online help for more
# information on querying this data from handlers.
#
# You can update WhitePages.txt file while the CIC system is running; changes
# to it will take effect shortly after the file is updated. The file consists
# of one entry per line; each entry consists of a partial (or full) telephone
# number followed by whitespace followed by address information. Specify a
# full address by separating lines with the pipe (|) character (the pipe is
# replaced with a newline character when the directory information is returned
# from the tool).
```

```
#
# All lookups in WhitePages.txt are done on a longest-matching leading string
# basis.
#
# ***Note: WhitePages.txt is not designed for large numbers of entries. We have
# developed and ship with at least one third-party interface for reverse white
# pages lookups (Redi-Data, Inc.); more may be added in future
# releases. In addition, the reverse white page driver COM interface is now
# open/public, which means customers or resellers are free to implement their
# own drivers in VB, C++, etc.
#
+1-844-274-5992|DisplayName: Genesys|Company: Genesys Telecommunications, Inc.|Address: 2001
Junipero Serra Blvd|City: Daly City|State: CA|Zip: 94014|Country: USA|
+1-800-555-8xxx|DisplayName: Genesys|Company: Genesys Telecommunications, Inc.|Address: 2001
Junipero Serra Blvd|City: Daly City|State: CA|Zip: 94014|Country: USA|
+1-954-698-0030|DisplayName: Genesys (Indianapolis)|Company: Genesys Telecommunications,
Inc.|Address: 7601 Interactive Way|Suite 201|City: Indianapolis|State: IN|Zip: 46278|Country:
USA|
+813.5510.8011|DisplayName: Genesys (Tokyo)|Company: Genesys Telecommunications, Inc.|Address:
10F Teikoku Hotel Tower|Room A-5 & 6|1-1-1 Uchisaiwai-cho|Chiyoda-ku|City: Tokyo|Zip: 100-
0011|Country: Japan|
+813.3593.2110|DisplayName: Genesys (Tokyo)|Company: Genesys Telecommunications, Inc.|Address:
10F Teikoku Hotel Tower|Room A-5 & 6|1-1-1 Uchisaiwai-cho|Chiyoda-ku|City: Tokyo|Zip: 100-
0011|Country: Japan|
```

Appendix D: Tracker Contact Resolution Prompt

When RWP lookup returns multiple RWP candidates, CIC client users are prompted to select the best matching name. This topic explains GUI features in the CIC client that prompt users to make a selection.

1. When RWP results are displayed to the user, a red line below the name indicates that user intervention is required.



2. Click on the prompt (underlined name) to proceed. A dropdown list will appear.



A list of candidate names is presented. These RWP candidates are generated by DataManager when it uses the Contact List sources configured in Interaction Administrator to perform lookups. If the user has a Tracker access license, additional options are presented, allowing the user to:

- Find Tracker Contact
 - Add Tracker Contact
3. Select a contact name from the list, or use a Tracker option to add or locate a contact.

Change Log

The following changes have been made to *Reverse White Page Lookups Technical Reference*, since Interaction Center version 4.0 product availability.

Date	Changes
01-August-2014	Updated 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 Interactive Intelligence product information site URLs, and copyright and trademark information.
01-July-2015	Updated cover page with new logo.
01-September-2015	Updated to reflect the addition of two CIC client applications, Interaction Desktop and Interaction Connect, and the removal of Interaction Client .NET edition. Updated old document formatting to better reflect documentation library standards.
25-April-2017	Updated to reflect the removal of Interaction Client Web Edition.
10-May-2018	Rebranded from Interactive Intelligence to Genesys.