



**Multimedia 7.6**

# **Selected Conceptual Data Models for the UCS Database**

**The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.**

Copyright © 2006–2008 Genesys Telecommunications Laboratories, Inc. All rights reserved.

## About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit [www.genesyslab.com](http://www.genesyslab.com) for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

## Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

## Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

## Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, [www.SoftwareRenovation.com](http://www.SoftwareRenovation.com).

## Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

## Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	<a href="mailto:support@genesyslab.com">support@genesyslab.com</a>
Europe, Middle East, and Africa	+44-(0)-118-974-7002	<a href="mailto:support@genesyslab.co.uk">support@genesyslab.co.uk</a>
Asia Pacific	+61-7-3368-6868	<a href="mailto:support@genesyslab.com.au">support@genesyslab.com.au</a>
Japan	+81-3-5649-6871	<a href="mailto:support@genesyslab.co.jp">support@genesyslab.co.jp</a>

**Prior to contacting technical support, please refer to the [Genesys Technical Support Guide](#) for complete contact information and procedures.**

## Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the [Genesys 7 Licensing Guide](#).

## Released by

Genesys Telecommunications Laboratories, Inc. [www.genesyslab.com](http://www.genesyslab.com)

**Document Version:** 76mm\_db\_11-2007\_v7.6.001.00



# Table of Contents

<b>Preface</b>	<b>5</b>
Multimedia and the CIM Platform .....	6
CIM Platform .....	6
Multimedia .....	6
Intended Audience.....	8
Chapter Summaries.....	8
Document Conventions .....	8
Related Resources .....	10
Making Comments on This Document .....	12
 <b>Chapter 1</b>	 <b>13</b>
<b>Introduction.....</b>	<b>13</b>
Overview.....	13
Data Types .....	13
Relationships .....	14
 <b>Chapter 2</b>	 <b>15</b>
<b>Contact Package.....</b>	<b>15</b>
Overview.....	15
Contact Entity .....	17
ContactAttributeMetaData Entity .....	19
ContactAttribute Entity.....	22
ContactArchiveSynchro .....	24
Sortable .....	26
 <b>Chapter 3</b>	 <b>27</b>
<b>ContactMerge Package .....</b>	<b>27</b>
Overview.....	27
InteractionMergeHistory Entity .....	28
MergeHistory Entity .....	29
Relationships Between ContactMerge and Contact .....	31
 <b>Chapter 4</b>	 <b>33</b>
<b>Interaction Package.....</b>	<b>33</b>
Overview.....	33
Interaction Entity.....	33

Making an Attached Attribute Sortable .....	41
EmailIn Entity.....	42
EmailOut Entity.....	45
<b>Index</b> .....	<b>49</b>



## Preface

Welcome to the *Multimedia 7.6 Selected Conceptual Data Models for the UCS Database*. This document describes those parts of the Universal Contact Server (UCS) database that are exposed to users for use in generating custom reports.

Multimedia was known as Multi-Channel Routing in releases 7.0 and 7.1. This document is valid for all 7.6 releases of Multimedia.

---

**Note:** For versions of this document created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library CD, which you can order by e-mail from Genesys Order Management at [orderman@genesyslab.com](mailto:orderman@genesyslab.com).

---

This preface provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

- [Multimedia and the CIM Platform, page 6](#)
- [Intended Audience, page 8](#)
- [Chapter Summaries, page 8](#)
- [Document Conventions, page 8](#)
- [Related Resources, page 10](#)
- [Making Comments on This Document, page 12](#)

---

# Multimedia and the CIM Platform

Genesys Multimedia (formerly Multi-Channel Routing) is a cover term for Genesys components that work together to manage interactions whose media is something other than traditional telephonic voice (for example, e-mail or chat).

Multimedia includes some parts of the Genesys Customer Interaction Management (CIM) Platform, plus certain of the media channels that run on top of the Platform.

## CIM Platform

The CIM Platform consists of the following:

- Management Framework
- Reporting (CC Analyzer, CCPulse+)
- Interaction Management, which in turn consists of:
  - Universal Routing
  - Interaction Workflow
  - Knowledge Management
  - Content Analysis
  - Universal Contact History

On top of the CIM Platform are various media channels. Some, such as Genesys Network Voice, handle traditional telephony. Others, such as Genesys E-mail, handle other media.

## Multimedia

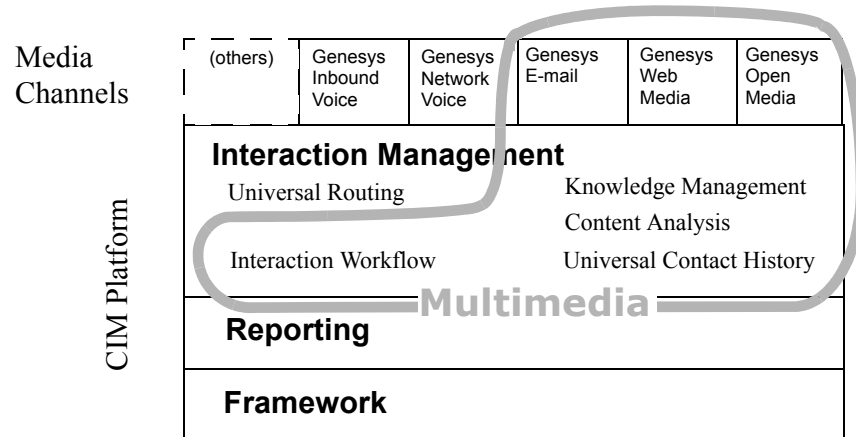
Multimedia, then, consists of the following:

- From the CIM Platform, all of Interaction Management except for Universal Routing:
  - Interaction Workflow—centralized handling of interactions irrespective of media type
  - Knowledge Management—creation and maintenance of standard responses and screening rules
  - Content Analysis—optional enhancement to Knowledge Management, applying natural language processing technology to categorize interactions
  - Universal Contact History—storage of data on contacts and on interactions (linked as threads)

Universal Routing is not considered part of Multimedia because it deals with both traditional telephonic interactions and the nontraditional interactions that are handled in Multimedia.

- From the media channels, at least one of the following:
  - Genesys E-mail—e-mail
  - Genesys Web Media—chat
  - Genesys Open Media—ability to add customized support for other media (fax, for example)
- Optionally, Web Collaboration—the ability for agents and customers to co-browse (simultaneously navigate) shared web pages. This is an option that you can add to either Genesys Web Media or Inbound Voice.

See [Figure 1](#).



**Figure 1: Multimedia in Relation to the CIM Platform and Media Channels**

**Note:** Although Universal Routing is not considered part of Multimedia, any functioning solution (platform plus channels) that includes any part of the Interaction Management sector requires Universal Routing.

## Licensing

Licensing requirements are:

- For each agent: one Multimedia Agent seat.
- For each media option: one media channel (E-mail and/or Web Media and/or custom media).
- For Genesys Content Analyzer: NLP Content Analysis license.

See also the *Genesys 7 Licensing Guide*.

## Reporting

Reporting templates are available for Multimedia. For details see the *Reporting Technical Reference Guide for the Genesys 7.x Release*.

---

## Intended Audience

This document, primarily intended for all users involved in setting up Genesys Multimedia, assumes that you have a basic understanding of:

- E-mail and web technology.
- Network design and operation.
- Your own network configurations.

You should also be familiar with:

- Genesys Framework architecture and functions.
- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.

---

## Chapter Summaries

In addition to this preface, this document contains the following chapters:

- Chapter 1, “Introduction,” on [page 13](#), defines terms and presents general information on how this document describes data types and relationships.
- Chapter 2, “Contact Package,” on [page 15](#), describes the entities in the Contact package.
- Chapter 4, “Interaction Package,” on [page 33](#), describes the entities in the Interaction package.

---

## Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

### Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

72fr\_ref\_09-2005\_v7.2.000.00

You will need this number when you are talking with Genesys Technical Support about this product.



## Type Styles

### Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

- Examples:**
- Please consult the *Genesys 7 Migration Guide* for more information.
  - *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
  - Do *not* use this value for this option.
  - The formula,  $x + 1 = 7$  where  $x$  stands for . . .

### Monospace Font

A monospace font, which looks like teletype or typewriter text, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

- Examples:**
- Select the Show variables on screen check box.
  - Click the Summation button.
  - In the Properties dialog box, enter the value for the host server in your environment.
  - In the Operand text box, enter your formula.
  - Click OK to exit the Properties dialog box.
  - The following table presents the complete set of error messages T-Server® distributes in EventError events.
  - If you select true for the inbound-bsns-calls option, all established inbound calls on a local agent are considered business calls.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

- Example:**
- Enter exit on the command line.

## Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from

installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

## Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

## Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

---

## Related Resources

Consult these additional resources as necessary:

- *Multimedia 7.6 Deployment Guide*, which describes deployment procedures for all Multimedia components.
- *Multimedia 7.6 Reference Manual*, which provides a reference listing of all configuration options and of field codes used in standard responses.
- *Multimedia 7.6 Interaction Models Reference Manual*, which presents a set of basic interaction models, showing the components involved and the requests and events (messaging) sent among them.
- *Multimedia 7.6 User's Guide*, which provides overall information and recommendations on the use and operation of Multimedia.
- *Multimedia 7.6 Universal Contact Server Manager Help*, which is a guide to the Universal Contact Server Manager user interface.
- *Multimedia 7.6 Knowledge Manager Help*, which is a guide to the Knowledge Manager user interface.
- *Multimedia 7.6 Web API Reference*, which is a Javadoc listing of classes, methods, fields, and constants of the Web API portion of the Web API Server component.

- *Multimedia 7.6 Web API Client Developer's Guide*, which describes the structure of the Web API, explains the Simple and Compound Samples, and describes procedures for customizing them.
- *Genesys Developer Program 7.6 Interaction SDK (Java) Developer's Guide*, which introduces you to essential concepts for developing an agent interaction application built on the Agent Interaction Layer (AIL) library.
- *Genesys Desktop 7.6 Deployment Guide*, which describes deployment procedures for the Genesys Desktop.
- *Genesys Desktop 7.6 Developer's Guide*, which describes customizing the Genesys Desktop.
- *Genesys Desktop 7.6 Agent Help*, which is a guide to the Genesys Agent Desktop.
- *Genesys Desktop 7.6 Supervisor's Help*, which is a guide to the Genesys Supervisor Desktop.
- *Universal Routing 7.6 Reference Manual*, which contains descriptions of all routing strategy objects, including those that are specific to Multimedia.
- *Universal Routing 7.2 Strategy Samples*, which describes the sample strategies supplied with Universal Routing.
- *Universal Routing 7.2 Business Process User's Guide*, which contains step-by-step instructions for using Interaction Routing Designer to design interaction workflows. It also describes the sample business processes supplied with Multimedia.
- *Universal Routing 7.6 Interaction Routing Designer Help*, which is a guide to Interaction Routing Designer, including the portion of it that designs interaction workflows and business processes for Multimedia.
- "Multimedia Log Events" in *Framework 7.6 Combined Log Events Help*, which is a comprehensive list and description of all events that may be recorded in logs.
- *Genesys Technical Publications Glossary*, which ships on the Genesys Documentation Library CD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- *Genesys 7 Migration Guide*, also on the Genesys Documentation Library CD, which contains a documented migration strategy for Genesys product releases 5.x and later. Contact Genesys Technical Support for additional information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at <http://genesyslab.com/support>.
- Documentation on the other three members of the Genesys Customer Interaction Platform: Universal Routing, Reporting, and Management Framework.

Information on supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys 7 Supported Operating Systems and Databases*
- *Genesys 7 Supported Media Interfaces*

Genesys product documentation is available on the:

- Genesys Technical Support website at <http://genesyslab.com/support>.
- Genesys Documentation Library CD, which you can order by e-mail from Genesys Order Management at [orderman@genesyslab.com](mailto:orderman@genesyslab.com).

---

## Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to [Techpubs.webadmin@genesyslab.com](mailto:Techpubs.webadmin@genesyslab.com).

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.



## Chapter

# 1

## Introduction

This chapter contains the following sections:

- [Overview, page 13](#)
- [Data Types, page 13](#)
- [Relationships, page 14](#)

---

## Overview

This document describes the Universal Contact Server (UCS) database in terms of packages, entities, attributes, and relationships.

An *entity* is an abstraction that corresponds to what is called a table in most relational database management applications. Entities contain attributes, corresponding to columns in a table. Related entities are grouped into packages.

---

## Data Types

Abbreviations used to indicate data type are listed in [Table 1](#).

**Table 1: Data Type Abbreviations**

Abbreviation	Meaning
Ax	Fixed length ASCII text, x characters long
BL	Boolean
BT	Byte (value between 0 and 255)
DT	Date/time

**Table 1: Data Type Abbreviations (Continued)**

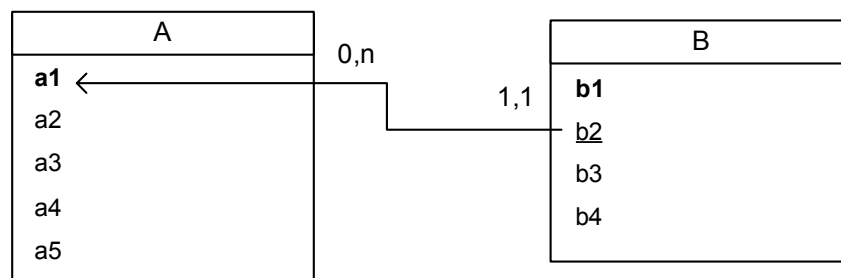
Abbreviation	Meaning
I	Integer
MBTxt	Multibyte text, unlimited size
BLOB	Binary large object, unlimited size
VMBT	Multibyte text, x characters long
SI	Short integer
VA	Variable ASCII text, x characters long

## Relationships

Relationships among entities are shown as in [Figure 2](#), where:

- Boxes A and B are entities
- Boldface indicates key fields.
- Underlining indicates foreign keys.
- Arrows point from foreign keys to the target field in the related entity.
- The notation  $E[x,y]$  indicates cardinality for relationships, meaning that between x and y records of the entity E must relate to the other entity in the relationship.

For example, the relationship in [Figure 2](#) between entity A [0,n] and entity B [1,1] is such that each A record relates to exactly one B record, and that each B record can relate to zero or more records of entity A.

**Figure 2: Relationship Between Entities**



## Chapter

# 2

## Contact Package

This chapter describes the `Contact` package, in the following sections:

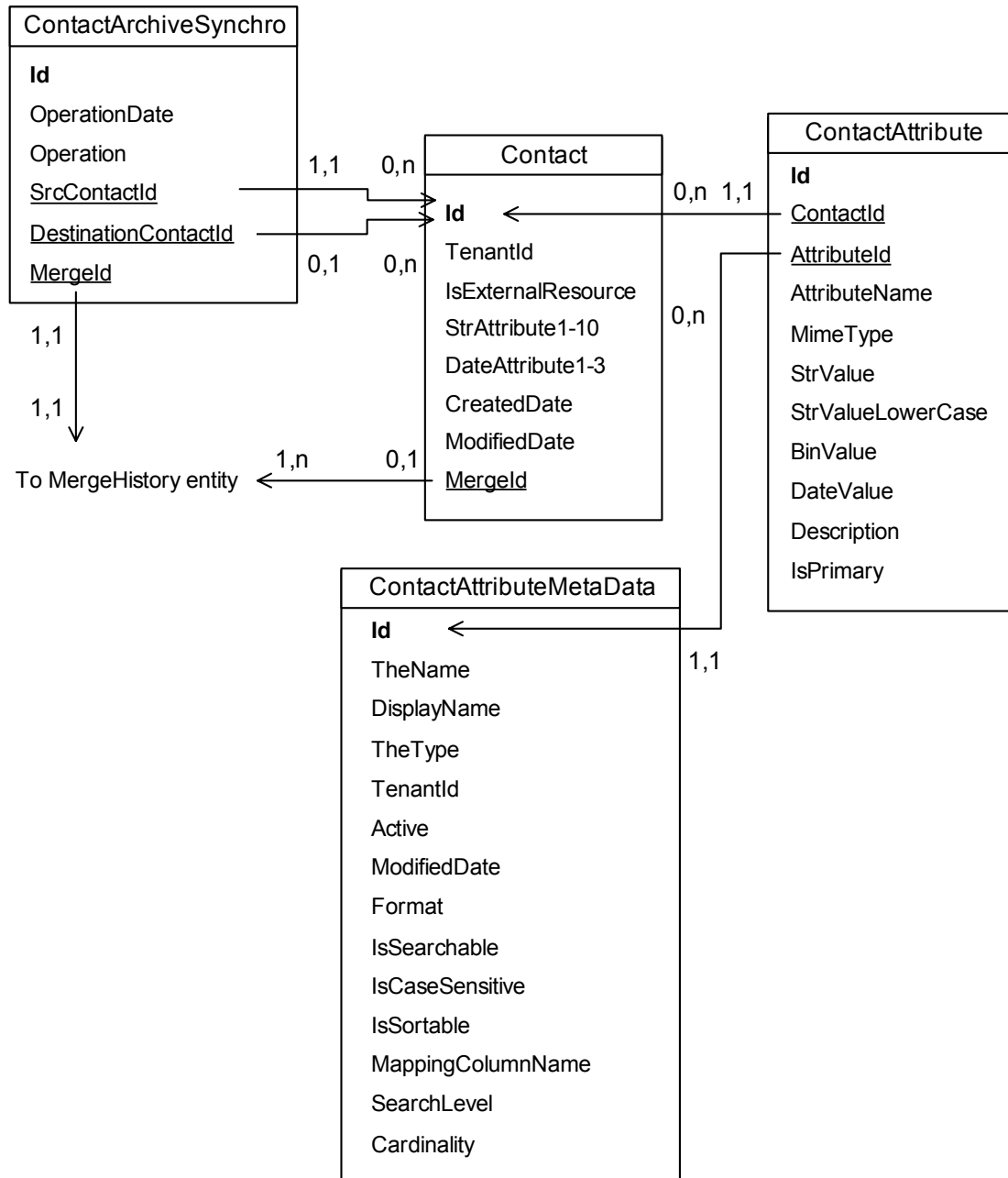
- [Overview, page 15](#)
- [Contact Entity, page 17](#)
- [ContactAttributeMetaData Entity, page 19](#)
- [ContactAttribute Entity, page 22](#)
- [ContactArchiveSynchro, page 24](#)
- [Sortable, page 26](#)

---

## Overview

The `Contact` package represents a contact center customer. It is closely related to the `ContactMerge` package, which keeps track of contact records that have been found to be duplicates and so merged with each other.

The `Contact` package consists of the `Contact`, `ContactAttribute`, `ContactAttributeMetaData`, and `ContactArchiveSynchro` entities. The relationships among these entities are shown in [Figure 3](#) (see “Relationships” on [page 14](#) for explanation of the notation).



**Figure 3: Contact Package Relationships**

The entities in the Contact package represent the following:

- The Contact entity defines contacts.
- The ContactAttributeMetaData entity defines attributes that a contact may have; these attributes correspond to Business Attributes in the Configuration Server database.



- The `ContactAttribute` entity provides a link between the `Contact` and `ContactAttributeMetaData` entities: it shows that a referenced contact actually has a referenced attribute, and it contains the value of that attribute.

For example, Contact A may have an e-mail address and a phone number, while Contact B has only an e-mail address. This is represented as follows:

- The `Contact` entity contains records `ContactA` and `ContactB`.
- `ContactAttributeMetaData` contains records `emailaddress` and `phonenummer`.
- `ContactAttribute` contains:
  - Record `X` that links `ContactA` with `emailaddress` and contains Contact A's e-mail address in its `StrValue` and `StrValueLowerCase` attributes.
  - Record `Y` that links `ContactA` with `phonenummer` and contains A's phone number in its `StrValue` and `StrValueLowerCase` attributes.
  - Record `Z` that links `ContactB` with `emailaddress` contains Contact B's e-mail address in its `StrValue` and `StrValueLowerCase` attributes.
- The `ContactArchiveSynchro` entity stores a record of operations on contacts. This entity replaces the 7.2 entity `MergedContact`.

The `Contact` package has many relationships with the `ContactMerge` package. For a summary of the relationships between these two packages, see Figure 5, "Relationships Between the `ContactMerge` and `Contact` Packages," on [page 32](#).

## Contact Entity

The `Contact` entity defines a contact and stores a few basic features of that contact.

The attributes of the `Contact` entity are shown in [Table 2](#).

**Table 2: Contact Entity Attributes**

Name	Type	Mandatory
Id (primary key)	VA16	Yes
TenantId	I	Yes
IsExternalResource	BL	No
StrAttribute1–10	VMBT256	No
DateAttribute1–3	VMBT256	No

**Table 2: Contact Entity Attributes (Continued)**

Name	Type	Mandatory
CreatedDate	DT	
ModifiedDate	DT	
MergeId	A16	No

Relationships of the Contact entity are shown in [Table 3](#).

**Table 3: Contact Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
ContactAttribute	ContactAttribute.ContactId, pointing to Contact.Id	Contact	1,1	0,n
Contact	Interaction.Id, pointing to Contact.Id	Interaction	0,n	0,1
MergeHistory	MergeHistory.SourceContactId, pointing to Contact.Id	Contact	0,1	0,n
MergeHistory	MergeHistory.DestinationContactId, pointing to Contact.Id	Contact	0,1	0,n
ContactArchiveSynchro	ContactArchiveSynchro.SrcContactId, pointing to Contact.Id	Contact	1,1	0,n
ContactArchiveSynchro	ContactArchiveSynchro.DestinationContact, pointing to Contact.Id	Contact	0,1	0,n
Contact	Contact.MergeId, pointing to MergeHistory.Id	MergeHistory	0,1	1,n

Following are descriptions of the attributes listed in [Table 2](#).

**Id**

Globally unique identifier of this contact.

**TenantId**

Tenant of this contact (database ID of the Tenant object in the Configuration Server database).

**IsExternalResource**

Indicates whether this contact is an external resource.

**StrAttribute1–10**

Stores duplicated contact attribute values marked as Sortable. See “Sortable” on [page 26](#).

**DateAttribute1–3**

Stores duplicated contact attribute values marked as Sortable. See “Sortable” on [page 26](#).

**CreatedDate**

Date and time that this contact record was created.

**ModifiedDate**

Date and time that this contact was last updated.

**Mergeld**

Stores the ID of the merge operation that created this contact. Foreign key to the MergeHistory entity, its target is MergeHistory.Id. A null value means that this contact is not the result of a merge operation.

---

## ContactAttributeMetaData Entity

This entity defines attributes that a contact can have. The inventory of these attributes, and some of their properties, is determined by the Contact Attributes objects located under Business Attributes in the Configuration Server database. When you change one of these Contact Attributes objects, the UCS database copies the values to the ContactAttributeMetaData entity.

Whether a given contact actually has one of these attributes, and if so what the value is, is shown by the ContactAttribute entity.

The attributes of the ContactAttributeMetaData entity are shown in [Table 4](#).

**Table 4: ContactAttributeMetaData Entity Attributes**

Name	Type	Mandatory
Id (primary key)	VA16	Yes
TheName	VMBT256	Yes
DisplayName	VMBT256	
TheType	BT	Yes
TenantId	I	Yes

**Table 4: ContactAttributeMetaData Entity Attributes (Continued)**

Name	Type	Mandatory
Active	BL	Yes
ModifiedDate	DT	Yes
Format	VMBT256	
IsSearchable	BL	Yes
IsCaseSensitive	BL	Yes
IsSortable	BL	Yes
MappingColumnName	VA20	
SearchLevel	BT	
Cardinality	BT	

Relationships of the ContactAttributeMetaData entity are shown in [Table 5](#)

**Table 5: ContactAttributeMetaData Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
ContactAttribute	AttributeId	ContactAttributeMetaData	1,1	0,n

**Id**

Unique identifier of this potential contact attribute.

**TheName**

System name of the attribute.

**DisplayName**

Display name of the attribute, as seen in Configuration Manager.

**TheType**

Type of the attribute, as follows:

- 0—string
- 1—date
- 2—binary

**TenantId**

ID of the tenant (in Configuration Server database) that this attribute is defined in.

**Active**

Defines whether this attribute is still present in Configuration Server.

**ModifiedDate**

Date that this attribute was last modified or created.

**Format**

Defines the format of this attribute.

**IsSearchable**

Indicates that values of this attribute can be used in the algorithm that creates new contact records. This algorithm does the following:

1. It checks whether the contact associated with a new interaction already exists in the database.
2. If the contact does not exist, it creates a new record to represent it.

To perform the action of Step 1, the algorithm searches for all contact records having a certain list of attributes and values. The `IsSearchable` attribute determines whether an attribute is on that list.

The default case is that the attributes `FirstName`, `LastName`, `Title`, `EmailAddress`, `PhoneNumber` have the value 1 for this attribute; all others have 0.

**IsCaseSensitive**

Defines whether this attribute is case-sensitive for lookup purposes.

**IsSortable**

Defines whether this attribute is sortable. See “Sortable” on [page 26](#).

**MappingColumnName**

Defines the column in the `Contact` entity in which this attribute will be duplicated if this attribute is marked as sortable. See “Sortable” on [page 26](#)

**SearchLevel**

Indicates the priority given to this attribute in searching (only relevant if this attribute’s `IsSearchable` attribute has the value 1). Default settings are as follows (0 is the highest priority, 127 the lowest):

EmailAddress—0

PhoneNumber—1

FirstName—2

LastName—2

all others—127

### Cardinality

Indicates the maximum number of distinct values for this attribute. For future use.

## ContactAttribute Entity

This entity indicates that a contact has one of the attributes listed in the `ContactAttributeMetaData` entity, and stores the value of the attribute.

The attributes of the `ContactAttribute` entity are shown in [Table 6](#).

**Table 6: ContactAttribute Entity Attributes**

Name	Type	Mandatory
Id (primary key)	VA16	Yes
ContactId	VA16	Yes
AttributeId	VA16	Yes
AttributeName	VMBT256	No
MimeType	VA256	No
StrValue	VMBT256	No
StrValueLowerCase	VMBT256	No
BinValue	BLOB	No
DateValue	DT	No
Description	VMBT256	No
IsPrimary	BL	Yes

Relationships of the `ContactAttribute` entity are shown in [Table 7](#).

**Table 7: ContactAttribute Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
ContactAttribute	ContactAttribute.ContactId, pointing to Contact.Id	Contact	1,1	0,n
ContactAttribute	ContactAttribute.AttributeId, pointing to ContactAttributeMetaData.Id	ContactAttributeMetaData	1,1	0,n

Following are descriptions of the attributes shown in [Table 6](#).

### **Id**

Unique identifier of this contact attribute.

### **ContactId**

The ID of the contact that this attribute value belongs to. Foreign key to `Contact` entity; its target is `Contact.Id`.

### **AttributeId**

The ID of the `ContactAttributeMetaData` record which describes this contact attribute. Foreign key to `ContactAttributeMetaData` entity; its target is `ContactAttributeMetaData.Id`.

### **AttributeName**

The Name of the `ContactAttributeMetaData` record that describes this contact attribute. Duplicates the value of `ContactAttributeMetaData.TheName`.

### **MimeType**

If this contact attribute has a binary value, defines its MIME type. Must comply with RFC 2046.

### **StrValue**

Stores the value of the linked `ContactAttributeMetaData` record if the latter is of type string.

### **StrValueLowerCase**

Stores the value of the linked `ContactAttributeMetaData` record, if the latter is of type string, but in lower case. This is used for case-insensitive queries.

**BinValue**

Stores the value of the linked `ContactAttributeMetaData` record if the latter is of type binary.

**DateValue**

Stores the value of the linked `ContactAttributeMetaData` record if the latter is of type date.

**Description**

Description of the value of the linked `ContactAttributeMetaData` record. This is useful in the case of attributes that have multiple values.

**IsPrimary**

Identifies the primary value of a contact attribute that has multiple values. For example, the database can contain two e-mail addresses for a single contact, but one must be marked as primary. In this case,

- The `ContactAttributeMetaData` entity contains a record defining the `email address` attribute.
- The `ContactAttribute` entity contains two records, both linked to `email address`, whose attributes include the following:
  - `Id = 0001`, `StrValue = joe_customer@home.org`, `IsPrimary = true`
  - `Id = 0002`, `StrValue = joe_customer@work.com`, `IsPrimary = false`

UCS enforces the following rule: if at least one value is defined for an attribute, then one and only one attribute value must be defined as primary.

---

## ContactArchiveSynchro

This entity stores a record of operations on `Contact` records. These operations, which include deleting, merging, and unmerging, take place only in the main database. The `ContactArchiveSynchro` entity provides a way to replicate these operations in the archive database.

The attributes of the `ContactArchiveSynchro` entity are shown in [Table 8](#).

**Table 8: ContactArchiveSynchro Entity Attributes**

Name	Type	Mandatory
Id (primary key)	A16	Yes
OperationDate	DT	Yes
Operation	SI	Yes



**Table 8: ContactArchiveSynchro Entity Attributes (Continued)**

Name	Type	Mandatory
SrcContactId	A16	Yes
DestinationContactId	A16	No
MergeId	A16	No

Relationships of the `ContactArchiveSynchro` entity are shown in [Table 9](#).

**Table 9: ContactArchiveSynchro Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
ContactArchiveSynchro	ContactArchiveSynchro.SrcContactId, pointing to Contact.Id	Contact	1,1	0,n
ContactArchiveSynchro	ContactArchiveSynchro.DestinationContactId, pointing to Contact.Id	Contact	0,1	0,n

Following are descriptions of the attributes listed in [Table 8](#).

**Id**

Unique identifier of this operation.

**OperationDate**

The date and time that this operation on the main database occurred.

**Operation**

Type of operation, as follows:

- 0—Delete
- 1—Merge
- 2—Unmerge

**SrcContactId**

Identifier of the source contact record. In a merge operation, source records are the duplicate records and the destination record is the one that merges the duplicates. Foreign key to the `Contact` entity; its target is `Contact.Id`.

**DestinationContactId**

Identifier of the destination contact record. Foreign key to the `Contact` entity; its target is `Contact.Id`.

**Mergeld**

Foreign key to the `MergeHistory` entity (see [page 29](#)); its target is `MergeHistory.Id`.

---

## Sortable

You can mark an attribute as sortable only if the `is-sortable` option on the Annex tab of the corresponding Business Attribute in the Configuration Server database is set to `true`. More specifically, the location of this option in Configuration Manager is `BusinessAttribute:ContactAttributes:AttributeValues:<attribute-name>:Annex:settings`.

If an attribute is marked as sortable, its primary value is copied to all new or updated `Contact` records that have that attribute, as a value of one of the `StrAttribute1–10` or `DateAttribute1–3` attributes (depending on the data type of the original attribute). This avoids the need for multiple queries to the `Contact` package when you want to retrieve contacts along with their attributes.

When you mark an attribute as sortable, the system copies its value to the (numerically) first unused `StrAttribute` or `DateAttribute`. That is, the first string attribute that you mark as sortable has its value copied to `StrAttribute1`, the second string attribute that you mark as sortable has its value copied to `StrAttribute2`, and so on. When all `StrAttribute` or `DateAttribute` attributes are used, the system ignores any further settings of `is-sortable = true`, and it displays a warning in the UCS log.



## Chapter

# 3

## ContactMerge Package

This chapter describes the `ContactMerge` package, in the following sections:

- [Overview, page 27](#)
- [InteractionMergeHistory Entity, page 28](#)
- [MergeHistory Entity, page 29](#)
- [Relationships Between ContactMerge and Contact, page 31](#)

---

### Overview

The `ContactMerge` package keeps track of contact records that have been found to be duplicates, and that therefore have been merged with each other. The merge operation is one of the fixes for data integrity issues that you can perform using UCS Manager (see *UCS Manager Help*).

The `ContactMerge` package consists of the `InteractionMergeHistory` and `MergeHistory` entities. `MergeHistory` represents the merge operation itself, and `InteractionMergeHistory` provides a link to the interactions that the merged contacts were associated with. The relationships among these entities are shown in [Figure 4](#) (see “Relationships” on [page 14](#) for explanation of the notation).

`ContactMerge` has many relationships with the `Contact` package. For a summary of the relationships between these two packages, see [Figure 5](#) on [page 32](#).

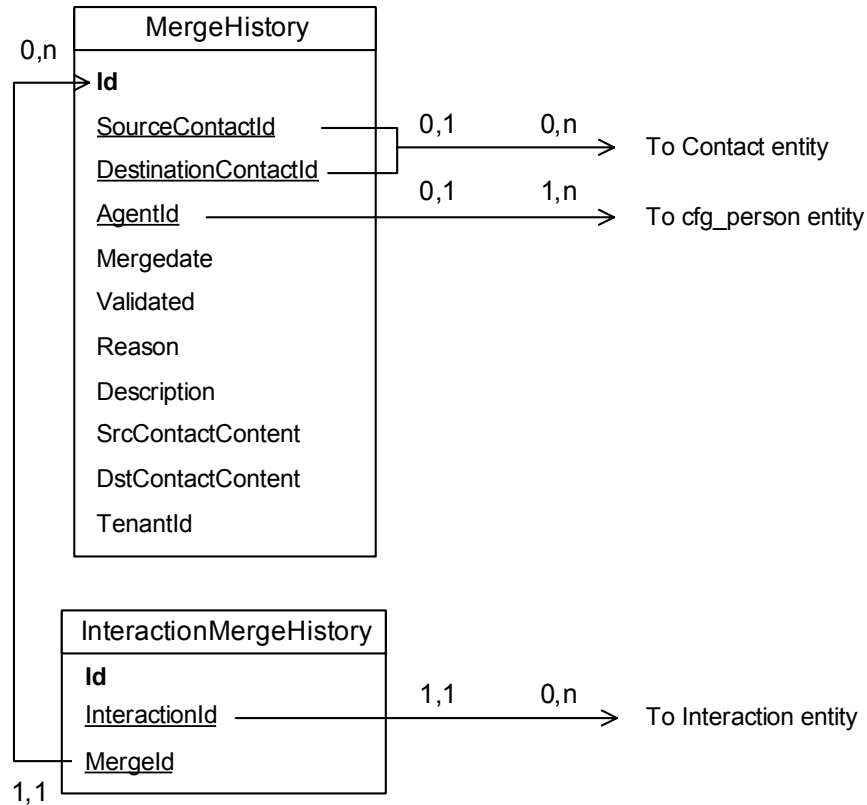


Figure 4: ContactMerge Package Relationships

## InteractionMergeHistory Entity

The **InteractionMergeHistory** entity stores a list of the interactions that are associated with a contact that has been merged into another contact. Storing this information makes it possible to undo a merge operation.

The attributes of the **InteractionMergeHistory** entity are shown in [Table 10](#).

Table 10: InteractionMergeHistory Entity Attributes

Name	Type	Mandatory
Id (primary key)	A16	Yes
MergeId	A16	Yes
InteractionId	BL	No

Relationships of the `InteractionMergeHistory` entity are shown in [Table 11](#).

**Table 11: InteractionMergeHistory Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
MergeHistory	InteractionMergeHistory.MergeId, pointing to MergeHistory.Id	InteractionMergeHistory	0,n	1,1
InteractionMergeHistory	InteractionMergeHistory.Interaction id, pointing to Interaction.Id	Interaction	1,1	0,n

Following are descriptions of the attributes listed in [Table 10](#).

### Id

Globally unique identifier of this merge history.

### MergeId

ID of the merge operation that created this merge history. Foreign key to `MergeHistory` entity; its target is `MergeHistory.Id`.

### InteractionId

ID of an interaction that the merged contact was associated with. Foreign key to the `Interaction` entity; its target is `Interaction.Id`.

## MergeHistory Entity

This entity represents a contact merge operation.

The attributes of the `MergeHistory` entity are shown in [Table 12](#).

**Table 12: MergeHistory Entity Attributes**

Name	Type	Mandatory
Id (primary key)	VA16	Yes
AgentId	I	No
TenantId	I	Yes
MergeDate	DT	Yes
SourceContactId	A16	Yes
DestinationContactId	A16	Yes

**Table 12: MergeHistory Entity Attributes (Continued)**

Name	Type	Mandatory
SrcContactContent	BLOB	Yes
DstContactContent	BLOB	Yes
Validated	BL	Yes
Reason	VMBT5	No
Description	VMBT5	No

Relationships of the MergeHistory entity are shown in [Table 13](#)

**Table 13: MergeHistory Entity Relationships**

Entity 1	Foreign Key	Entity 2	Cardinality	
			1 > 2	2 > 1
Contact	Contact.MergeId, pointing to MergeHistory.Id	MergeHistory	0,1	1,n
MergeHistory	InteractionMergeHistory.MergeId, pointing to MergeHistory.Id	InteractionMergeHistory	0,n	1,1
MergeHistory	MergeHistory.AgentId, pointing to cfg_person.dbid	cfg_person	0,1	1,n
MergeHistory	MergeHistory.SourceContactId, pointing to Contact.Id	Contact	0,1	0,n
MergeHistory	MergeHistory.DestinationContact.Id, pointing to Contact.Id	Contact	0,1	0,n

Following are descriptions of the attributes shown in [Table 12](#).

**Id**

Unique identifier of this merge operation.

**AgentId**

ID of the person who performed the merge operation. Must be an agent defined in the Configuration Layer. Foreign key to the `cfg_person` entity in the Configuration Layer database; its target is `cfg_person.dbid`.

**TenantId**

ID of the tenant in which this merge operation took place.

**MergeDate**

Date and time of this merge operation.

**SourceContactId**

ID of the source contact; that is, one of the contacts that was merged in this operation. Foreign key to the `Contact` entity; its target is `Contact.Id`

**DestinationContactId**

ID of the destination contact; that is, the contact into which the source contacts were merged in this operation

**SrcContactContent**

Content of the source contact before the merge operation.

**DstContactContent**

Content of the destination contact before the merge operation.

**Validated**

Reserved for future use.

**Reason**

Reason for the merge operation.

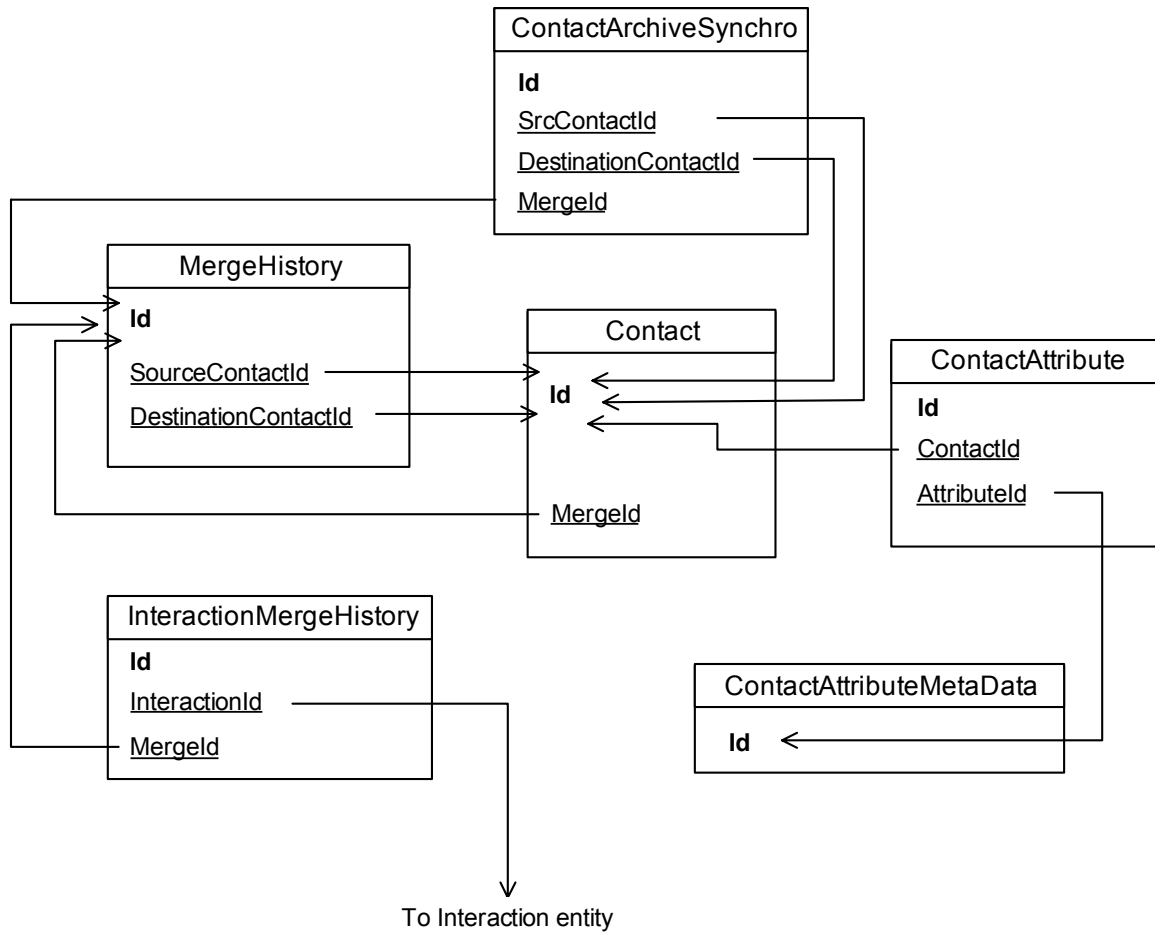
**Description**

Description of the merge operation.

---

## Relationships Between ContactMerge and Contact

The `Contact` and `ContactMerge` entities are closely related, with many relationships in both directions. [Figure 5](#) summarizes these relationships, displaying only the primary key and foreign keys of each entity. For the complete list of attributes and the relationship cardinalities, see [Figure 4](#), “ContactMerge Package Relationships,” on [page 28](#) and [Figure 3](#), “Contact Package Relationships,” on [page 16](#).



**Figure 5: Relationships Between the ContactMerge and Contact Packages**





## Chapter

# 4

## Interaction Package

This chapter contains the following sections:

- [Overview, page 33](#)
- [Interaction Entity, page 33](#)
- [EmailIn Entity, page 42](#)
- [EmailOut Entity, page 45](#)

---

### Overview

The `Interaction` package stores interactions. It consists of the `Interaction`, `EmailIn`, and `EmailOut` entities.

The `Interaction` entity stores all interactions handled by the system, with all the common attributes that any interaction can have. Each particular type of interaction (inbound or outbound e-mail, chat, phone call, and so on), is also represented in a separate entity for that type. For example, each incoming e-mail interaction has one record in the `Interaction` entity and another in the `EmailIn` entity, linked by the relationship `Interaction.Id = EmailIn.Id`.

This document describes only selected entities related to e-mail. Entities representing other media are not described here.

---

### Interaction Entity

The `Interaction` entity stores the attributes that are common to all interaction types.

The attributes of the Interaction entity are shown in [Table 14](#). See “Data Types” on [page 13](#) for data type abbreviations.

**Table 14: Interaction Entity Attributes**

Name	Type	Mandatory
Id (primary key)	VA16	Yes
Status	BT	Yes
EntityTypeId	BT	Yes
MediaTypeId	VMBT32	Yes
TypeId	VMBT32	Yes
SubtypeId	VMBT32	No
ExternalId	VMBT256	No
OwnerId	I	No
ContactId	VA16	No
ParentId	VA16	No
StartDate	DT	Yes
EndDate	DT	No
ThreadId	VA16	No
CategoryId	VA16	No
Timeshift	SI	No
Subject	VMBT512	No
Text	MBTxT	No
StructuredText	MBTxT	No
StructTextMimeType	VA256	No
TheComment	MBTxT	No
TenantId	I	Yes
ThreadHash	I	No
CanBeParent	BL	Yes
CreatorAppId	I	Yes

**Table 14: Interaction Entity Attributes (Continued)**

Name	Type	Mandatory
QueueName	VMBT64	No
AllAttributes	PIC64000	No
StrAttribute1–StrAttribute10	VMBT256	No
IntAttribute1– IntAttribute5	I	No
IsSpam	BL	No
WebSafeEmailStatus	VMBT32	No
IsCategoryApproved	BL	No
StoppedReason	VMBT64	No
Lang	VA64	No

Relationships of the Interaction entity are shown in [Table 15](#).

**Table 15: Interaction Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
Interaction	Interaction.ContactId, pointing to Contact.Id	Contact	1,1	0,n
Interaction	Interaction.ParentId, pointing to Interaction.Id	Interaction	1,1	0,n

Following are descriptions of the attributes listed in [Table 14](#).

**Id**

The identifier of this interaction.

**Status**

The status of the interaction. The value 2 means that the interaction is being processed by Interaction Server; all other values mean that it is not. Possible values are as follows:

0—New. The interaction has been created and is ready for pre-processing (specific to e-mails).

1—Pending. The interaction is waiting for submission to Interaction Server (specific to e-mails).

- 2—In process. The interaction is currently being handled by Interaction Server.
- 3—Stopped. Processing of the interaction has ended.

**EntityType**

The type of database entity used for the interaction (in addition to the Interaction entity). This is different from MediaType, which is an interaction attribute. Possible values are as follows:

- 0—EmailIn
- 1—EmailOut
- 2—Chat
- 3—PhoneCall
- 5—CallBack
- 6—CoBrowse
- 7—Generic interaction

**MediaType**

The media type of the interaction. This attribute can have as its value anything defined in the Configuration Server database as a Business Attributes > Media Type object; for example:

- callback
- chat
- cobrowsing
- email
- voice

**Type**

This attribute can have as its value anything defined in the Configuration Server database as a Business Attributes > Interaction Type object. The default values that are created when you install Configuration Server are as follows:

- Inbound
- Internal
- Outbound

**Subtype**

The subtype of the interaction. This attribute can have as its value anything defined in the Configuration Server database as a Business Attributes > Interaction Subtype object. The default values that are created when you install Configuration Server are as follows:

**InboundCollaborationReply**—Reply from an external resource that has been invited to join a collaboration session.

**InboundCustomerReply**—Reply to a previous known interaction.

**InboundNDR**—Non-delivery report.

**InboundNew**—The e-mail is newly arrived in the system.

**InternalCollaborationInvite**—Invitation to a collaboration session, issued to an internal resource.

**InternalCollaborationReply**—Reply from an internal resource that has been invited to join a collaboration session.

**OutboundAcknowledgement**—Acknowledgement sent to the customer, usually automated.

**OutboundAutoResponse**—Automated reply to the customer.

**OutboundCollaborationInvite**—Invitation to a collaboration session, issued to an external resource.

**OutboundNew**—Newly-initiated outbound email, not a reply to an inbound interaction.

**OutboundRedirect**—Redirected outbound e-mail for an inbound interaction. The final message will appear as if it was sent directly from the customer.

**OutboundReply**—Manual reply to an inbound interaction.

### **ExternalId**

Stores a reference identifier that can be used or set by an external system; (for example, for threading). For e-mail it duplicates the header field `Message-Id`.

### **OwnerId**

The identifier of the person who owns the interaction. If the interaction is in the `Stopped` state, the identifier of the most recent owner of the interaction. It contains the Configuration Server database ID of a `Person` object.

### **ContactId**

The identifier of the related `Contact` entity, if any. Can be null, as in the case of an agents-only chat session. Foreign key to `Contact` entity; its target is `Contact.Id`.

### **ParentId**

The identifier of the parent interaction; used for interaction threading. If the interaction has no parent, it is the root of a thread. Foreign key to another record in the `Interaction` entity; its target is `Interaction.Id`.

**StartDate**

The date and time, in UTC, when the interaction was created in the system; that is, the moment the record was created in the UCS database.

**EndDate**

The date and time, in UTC, when processing of the interaction stopped; that is, when `Interaction.Status` is set to 3 (Stopped).

**ThreadId**

Identifies the thread to which the interaction belongs. A new identifier must be used for this attribute each time a new root interaction is created. Child interactions must inherit this attribute value.

**CategoryId**

The identifier of the category assigned to the interaction by Content Analysis. See also “`IsCategoryApproved`” on [page 40](#).

**Timeshift**

Specifies how many minutes must be added or subtracted from UTC time to get the local time of the location to which the interaction is related.

**Subject**

Contains the subject of the interaction. Can be used to display the interaction in a tree. If the interaction is a MIME-encoded e-mail, the string stored in this attribute must be decoded. The source of this attribute is as follows:

- E-mail—Subject
- Chat—Chat session summary
- Phone call—Entered by agent

Although RFC2822 does not state a maximum size for header fields, the maximum length for this UCS database attribute is 512 characters. This is to preserve efficiency in searching.

**Text**

Contains the displayable text of the interaction, without any structure or format information. For e-mails this is the plain text body (if any). For chat it is the plain text version of the transcript.

This data type does not allow lookup.

**StructuredText**

Contains the structured text of the interaction. For e-mail whose body is HTML formatted, it is the HTML body. For chat it is a formatted version of the transcript.

This data type does not allow lookup.

**StructTextMimeType**

MIME type of the [StructuredText](#) attribute; must comply with RFC 2046.

**TheComment**

Comment related to the interaction, usually entered by an agent.

**TenantId**

Identifier of the tenant with which the interaction is associated; the primary identifier of the foreign `ConfigSvr.Tenant` entity. It must be a tenant declared in Configuration Server.

**ThreadHash**

Hashcode computed from the `Subject` attribute. It enables the system to thread by subject.

**CanBeParent**

Indicates if this thread is closed or open. With a value of `0` (or `no` or `false`), the interaction cannot have any child, so the thread is closed. With a value of `1` (or `yes` or `true`), the interaction can have a child, so the thread remains open. In the case of a complex thread, with multiple branches, you can close the entire thread by updating this attribute in all the interactions of the thread.

**CreatorAppId**

Identifier of the application that created the interaction. This provides a recovery process for synchronous interactions. If a synchronous interaction is in an unclosed state after a crash or a server disconnection, UCS can discover from this attribute which application it was that created the interaction.

If no `CreatorAppId` is available, UCS uses its own Application ID (database ID of its Application object in Configuration Server) instead.

**QueueName**

The name of the queue being used to submit the interaction to Interaction Server.

**AllAttributes**

Stores all attributes attached to an Interaction entity. Interaction attributes can be strings, integers, or attribute lists (recursive).

`AllAttributes` is for storage only. To perform lookups on an attached attribute, you must use one of the following two specialized attributes.

**StrAttribute1–StrAttribute10**

Allows you to perform read and lookup on a specific attached attribute containing a string. You must map the desired attribute onto an `InteractionAttribute` Business Attribute object in the Configuration Server database.

For more details, see “Making an Attached Attribute Sortable” on [page 41](#).

**IntAttribute1–IntAttribute5**

Allows you to perform read and lookup on a specific attached attribute containing an integer. You must map the desired attribute onto an `InteractionAttribute` Business Attribute object in the Configuration Server database.

For more details, see “Making an Attached Attribute Sortable” on [page 41](#).

**IsSpam**

Applies to `EmailIn`; specifies whether this interaction is considered to be spam. Value set using supervisor desktop. Possible values:

- 0—not spam
- 1—spam

**WebSafeEmailStatus**

Reserved for later use.

**IsCategoryApproved**

Allows an agent to approve or disapprove the category that was suggested by Classification Server. Used as the feedback mechanism for model evaluation and classification training. Possible values:

- False—Disapproved; the agent used manual override to change the suggested category.
- True—Approved; the agent confirms the suggested category.
- Null—No feedback from agent.

**StoppedReason**

The reason why the interaction has stopped. Each stop object in a routing strategy is expected to provide a reason.



**Lang**

The language of the interaction.

**Making an Attached Attribute Sortable**

By default, all attributes that are attached to an interaction are stored in the `AllAttributes` attribute of the `Interaction` entity. You can make sortable any attribute that is represented in the Configuration Server database by a `BusinessAttributes` object of type `InteractionAttributes`. Examples are:

- Category
- Disposition Code
- Interaction Subtype
- Interaction Type
- Language
- Media Type
- Priority
- Reason Code
- Service Type
- StopProcessing Reason

The `Interaction` entity includes attributes `StrAttribute1–StrAttribute10` and `IntAttribute1–IntAttribute5`, which exist to enable you to make attached attributes searchable. These attributes `StrAttribute1–StrAttribute10` and `IntAttribute1–IntAttribute5` may be referred to collectively as *replicant attributes*, as explained below.

To be able to perform lookup on any of the attached attributes, you must do the following in Configuration Manager:

3. Be sure that Properties windows show their Annex tabs. If they do not:
  - a. Go to the `View` menu and select `Options`.
  - b. In the resulting dialog box, select the `Show Annex tab in object properties` check box.
4. In the tenant for your UCS, go to `Business Attributes > Interaction Attributes > Attribute Values`.
5. Open the Properties window for the attribute that you want to make searchable (for example, `Service Type`).
6. On the Annex tab, create a section named `settings` if it does not already exist.
7. In this `settings` section, create an option named `is-sortable` and give it the value `true`.

8. If the attribute is of type string, you are finished. If it is of type integer, you must create an additional option, also in the `Settings` section, named `type` with the value `integer`.

Once you have configured an attached attribute as sortable, UCS takes its value as stored in `AllAttributes` and copies it as the value of one of the replicant attributes. To find out which replicant attribute copies a given attached attribute, look at the content of the `IxnAttributeMetaData` table. For example, if you have configured the `ServiceType` attribute to be searchable, you can find out which replicant attribute copies its value by using the following SQL request:

```
select MappingColumnName from IxnAttributeMetaData where
TheName='ServiceType';
```

Please also note the following:

- This replication process only applies to interactions created or updated after you perform the configuration described in this section. The replication process is not applied to interactions retroactively. The replicant attribute in older records will remain empty.
- By default, the replicant attribute that replicates the Interaction attribute does not have any database index. To increase performance during queries, consider adding index(es) to those replicant attributes that are in use.
- Replicant attributes are read-only from outside UCS. UCS is responsible for synchronization of their content whenever `Interaction.AllAttributes` is updated.
- The mapping between a searchable interaction attribute and a replicant attribute is based on the type (string or integer) of the business attribute declared in the Configuration Server database (string by default). UCS chooses from among the replicant attributes of the proper type that are not already associated with an attached attribute. It does this until no more replicant attributes are available.
- Once a replicant attribute has been used for a particular attribute, it is dedicated to that attribute: it cannot be used for another one. The only modification you can make is to configure a searchable attribute to be no longer searchable. The replicant attribute that copied this attribute's values will then retain those values for existing records and for any new records.

---

## EmailIn Entity

The `EmailIn` entity contains all the attributes specific to incoming e-mails. The incoming e-mail can be of any of the following types: new e-mail from a customer (from a POP/IMAP account or from a web form), customer reply, reply from external agent, or non-delivery report.

The attributes of the `EmailIn` entity are shown in [Table 16](#).

**Table 16: EmailIn Entity Attributes**

Name	Type	Key	Mandatory
Id	VA16	Primary	Yes
FromAddress	VMBT256		No
FromPersonal	VMBT256		No
ReplyToAddress	VMBT256		No
ToAddresses	MBTXT		No
CcAddresses	MBTXT		No
BccAddresses	MBTXT		No
SentDate	DT		No
Mailbox	VMBT256		Yes
WhichRuleMatched	VA16		No
EmailOutId	VA16		No
Header	BLOB		No

Relationships of the `EmailIn` entity are shown in [Table 17](#).

**Table 17: EmailIn Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
EmailIn	EmailIn.Id, pointing to Interaction.Id	Interaction	1,1	0,n

Following are descriptions of the attributes listed in [Table 16](#).

### Id

The identifier of this incoming e-mail. Foreign key to `Interaction`; its target is `Interaction.Id`.

### FromAddress

The e-mail address of the sender of this e-mail. Directly related to the `From` header field. Contains the `addr-spec` part of the full e-mail address.

**FromPersonal**

The (optional) personal part of the `From` e-mail address (usually a displayable, fully localized user name). Stored fully decoded (from MIME encoding) in this entity.

**ReplyToAddress**

Contains the address to which a reply to the interaction should be sent. Directly related to the `Reply-To` header field. Contains only the `addr-spec` part of the full e-mail address.

---

**Note:** RFC 822 specifies that the `Reply-To` header field can include a list of e-mail addresses. However, if that field does contain a list of addresses, this entity stores only the first of them.

---

**ToAddresses**

The list of recipients for this e-mail interaction. Directly relates to the `To` header field. These addresses are stored as MIME-encoded to avoid parsing problems. No search ability is available on this field, due to MIME encoding and Character Large Object (CLOB) storage.

**CcAddresses**

The list of recipients to which copies of this e-mail interaction should be sent. Directly relates to the `cc` header field. These addresses are stored as MIME-encoded to avoid parsing problems. No search ability is available on this field, due to MIME encoding and CLOB storage.

**BccAddresses**

The list of recipients to send blind copies of this e-mail interaction to (these addresses do not appear in the outgoing e-mails). There is no related header field. These addresses are stored as MIME encoded to avoid parsing problems. No search ability is available on this field, due to Mime encoding and CLOB storage.

**SentDate**

The date and time, in Coordinated Universal Time (UTC) format, that the e-mail was sent. Directly relates to the `Date` header field of the MIME message. Decoding of this header field includes converting local time to UTC time.

**Mailbox**

Contains the e-mail address or account of the mailbox through which this inbound e-mail was retrieved.

**WhichRuleMatched**

Always null; reserved for future use.

**EmailOutId**

Refers to the child outbound e-mail, if any. This attribute is a shortcut to the child EmailOut.

**Header**

Contains the full text of the header of this inbound e-mail. Some other attributes, such as `FromAddress` and `ToAddresses`, have values that are extracted from the header, whereas `Header` contains the entire header text.

---

## EmailOut Entity

The `EmailOut` entity contains all the attributes specific to outgoing e-mail. The outgoing e-mail can be of any of the following types: new e-mail to a customer, manual reply from to a customer, acknowledgement to a customer, autoresponse to a customer, message redirected to another e-mail account, or e-mail sent to an external agent.

The attributes of the `EmailOut` entity are shown in [Table 18](#).

**Table 18: EmailOut Entity Attributes**

Name	Type	Key	Mandatory
Id	VA16	Primary	Yes
FromAddress	VMBT256		No
FromPersonal	VMBT256		No
ReplyToAddress	VMBT256		No
ToAddresses	MBTXT		No
CcAddresses	MBTXT		No
BccAddresses	MBTXT		No
SentDate	DT		No
Forward	I		Yes
ReferenceId	VA255		No

**Table 18: EmailOut Entity Attributes (Continued)**

Name	Type	Key	Mandatory
ReviewerId	I		No
StandardResponseId	VA16		No

Relationships of the EmailOut entity are shown in [Table 17](#).

**Table 19: EmailOut Entity Relationships**

Entity 1	Foreign key	Entity 2	Cardinality	
			1 > 2	2 > 1
EmailOut	EmailOut.Id, pointing to Interaction.Id	Interaction	1,1	0,n

Following are descriptions of the attributes listed in [Table 18](#).

**Id**

The identifier of this outgoing e-mail. Foreign key to Interaction; its target is Interaction.Id.

**FromAddress**

The e-mail address of the sender of this e-mail interaction. Directly related to the From header field. Contains the addr-spec part of the full e-mail address.

**FromPersonal**

The (optional) personal part of the From e-mail address (usually a displayable, fully localized user name). Stored fully decoded (from MIME encoding) in this entity.

**ReplyToAddress**

Contains the address to which a reply to the interaction should be sent. Directly related to the Reply-To header field. Contains only the addr-spec part of the full e-mail address.

**ToAddresses**

The list of recipients for this e-mail interaction. Directly relates to the To header field. These addresses are stored as MIME-encoded to avoid parsing problems. No search ability is available on this field, due to MIME encoding and Character Large Object (CLOB) storage.

**CcAddresses**

The list of recipients to which copies of this e-mail interaction should be sent. Directly relates to the `cc` header field. These addresses are stored as MIME-encoded to avoid parsing problems. No search ability is available on this field, due to MIME encoding and CLOB storage.

**BccAddresses**

The list of recipients to which blind copies of this e-mail interaction should be sent (these addresses do not appear in the outgoing e-mails). There is no related header field. These addresses are stored as MIME-encoded to avoid parsing problems. No search ability is available on this field, due to MIME encoding and CLOB storage.

**SentDate**

The date and time, in Coordinated Universal Time (UTC) format, that the e-mail was sent. Directly relates to the `Date` header field of the MIME message. Decoding of this header field includes converting local time to UTC time.

**Forward**

Reserved for future use. The value is always 0.

**Referenceld**

Contains the external identifier (`Message-Id`) of the parent e-mail. This is for optimization purposes: when a new e-mail is generated from this record, this attribute is used to build the `In-Reply-To` header. This avoids the need to go load the parent interaction record from the database.

**ReviewerId**

Contains the Configuration Server database ID of the Person (agent) who reviewed this EmailOut.

**StandardResponseld**

Reserved for future use.







# Index

## A

Active	
attribute of ContactAttributeMetaData . . . . .	20, 21
AgentId	
attribute of MergeHistory . . . . .	29, 30
AllAttributes	
attribute of Interaction . . . . .	35, 40
attribute	
defined. . . . .	13
marking as sortable . . . . .	26
AttributId	
attribute of ContactAttribute . . . . .	22, 23
AttributeName	
attribute of ContactAttribute . . . . .	22, 23
audience	
defining . . . . .	8
Ax . . . . .	13

## B

BccAddresses	
attribute of EmailIn . . . . .	43, 44
attribute of EmailOut. . . . .	45, 47
BinValue	
attribute of ContactAttribute . . . . .	22, 24
BL . . . . .	13
BLOB. . . . .	14
BT . . . . .	13
Business Attributes. . . . .	41

## C

CanBeParent	
attribute of Interaction . . . . .	34, 39
Cardinality	
attribute of ContactAttributeMetaData . . . . .	20, 22
cardinality	
defined. . . . .	14
CategoryId	

attribute of Interaction. . . . .	34, 38
CcAddresses	
attribute of EmailIn . . . . .	43, 44
attribute of EmailOut . . . . .	45, 47
chapter summaries	
defining . . . . .	8
CIM (Customer Interaction Management)	
Platform . . . . .	6
commenting on this document. . . . .	12
Contact entity . . . . .	17–19
relations with ContactMerge entity summarized	31
Contact package. . . . .	15–26
ContactArchiveSynchro entity . . . . .	24–26
ContactAttribute entity . . . . .	22–24
ContactAttributeMetaData entity. . . . .	19–22
ContactId	
attribute of ContactAttribute. . . . .	22, 23
attribute of Interaction. . . . .	34, 37
ContactMerge package . . . . .	27–32
CreatedDate	
attribute of Contact . . . . .	18, 19
CreatorAppld	
attribute of Interaction. . . . .	34, 39

## D

data type. . . . .	13
DateAttribute	
attribute of Contact . . . . .	17, 19
DateValue	
attribute of ContactAttribute. . . . .	22, 24
Description	
attribute of ContactAttribute. . . . .	22, 24
attribute of MergeHistory . . . . .	30, 31
DestinationContactId	
attribute of ContactArchiveSynchro. . . . .	25, 26
attribute of MergeHistory . . . . .	29, 31
DisplayName	
attribute of ContactAttributeMetaData . . . . .	19, 20
document	

conventions . . . . . 8  
 errors, commenting on . . . . . 12  
 version number . . . . . 8  
 DstContactContent  
   attribute of MergeHistory . . . . . 30, 31  
 DT . . . . . 13

## E

EmailIn entity . . . . . 42–45  
 EmailOut entity . . . . . 45–47  
 EmailOutId  
   attribute of EmailIn . . . . . 43, 45  
 EndDate  
   attribute of Interaction . . . . . 34, 38  
 entity  
   defined. . . . . 13  
 EntityTypeId  
   attribute of Interaction . . . . . 34, 36  
 ExternalId  
   attribute of Interaction . . . . . 34, 37

## F

Format  
   attribute of ContactAttributeMetaData . . 20, 21  
 Forward  
   attribute of EmailOut. . . . . 45, 47  
 FromAddress  
   attribute of EmailIn . . . . . 43  
   attribute of EmailOut. . . . . 45, 46  
 FromPersonal  
   attribute of EmailIn . . . . . 43, 44  
   attribute of EmailOut. . . . . 45, 46

## H

Header  
   attribute of EmailIn . . . . . 43, 45

## I

I  
   data type abbreviation. . . . . 14  
 Id  
   attribute of Contact . . . . . 17, 18, 28, 29  
   attribute of ContactArchiveSynchro . . . 24, 25  
   attribute of ContactAttribute . . . . . 22, 23  
   attribute of ContactAttributeMetaData . . 19, 20  
   attribute of EmailIn . . . . . 43  
   attribute of EmailOut. . . . . 45, 46  
   attribute of Interaction . . . . . 34, 35  
   attribute of MergeHistory . . . . . 29  
 IntAttribute

  attribute of Interaction. . . . . 35, 40  
 Interaction  
   entity . . . . . 33–42  
   package. . . . . 33–47  
 InteractionId  
   attribute of InteractionMergeHistory . . 28, 29  
 InteractionMergeHistory entity . . . . . 28–29  
 IsCaseSensitive  
   attribute of ContactAttributeMetaData . . 20, 21  
 IsCategoryApproved  
   attribute of Interaction. . . . . 35, 40  
 IsExternalResource  
   attribute of Contact . . . . . 17, 18  
 IsPrimary  
   attribute of ContactAttribute. . . . . 22, 24  
 IsSearchable  
   attribute of ContactAttributeMetaData . . 20, 21  
 IsSortable  
   attribute of ContactAttributeMetaData . . 20, 21  
 IsSpam  
   attribute of Interaction. . . . . 35, 40

## L

Lang  
   attribute of Interaction. . . . . 35, 41

## M

Mailbox  
   attribute of EmailIn . . . . . 43, 44  
 MappingColumnName  
   attribute of ContactAttributeMetaData . . 20, 21  
 MBTxt . . . . . 14  
 media channels . . . . . 6  
 MediaTypeId  
   attribute of Interaction. . . . . 34, 36  
 MergeDate  
   attribute of MergeHistory . . . . . 29, 31  
 MergedContact  
   entity in 7.2 . . . . . 17  
 MergeHistory entity . . . . . 29–31  
   relations with Contact entity summarized . . 31  
 Mergeld  
   attribute of Contact . . . . . 18, 19  
   attribute of ContactArchiveSynchro . . . 25, 26  
   attribute of InteractionMergeHistory . . . 28, 29  
 MimeType  
   attribute of ContactAttribute. . . . . 22, 23  
 ModifiedDate  
   attribute of Contact . . . . . 18, 19  
   attribute of ContactAttributeMetaData . . 20, 21

**O**

Operation  
 attribute of ContactArchiveSynchro . . . 24, 25  
 OperationDate  
 attribute of ContactArchiveSynchro . . . 24, 25  
 OwnerId  
 attribute of Interaction . . . 34, 37

**P**

package  
 defined. . . . . 13  
 ParentId  
 attribute of Interaction . . . 34, 37

**Q**

QueueName  
 attribute of Interaction . . . 35, 39

**R**

Reason  
 attribute of MergeHistory . . . 30, 31  
 Referenceld  
 attribute of EmailOut. . . . . 45, 47  
 relationships  
 in diagrams . . . . . 14  
 ReplyToAddress  
 attribute of EmailIn . . . . . 43, 44  
 attribute of EmailOut. . . . . 45, 46  
 ReviewerId  
 attribute of EmailOut. . . . . 46, 47

**S**

SearchLevel  
 attribute of ContactAttributeMetaData . . 20, 21  
 SentDate  
 attribute of EmailIn . . . . . 43, 44  
 attribute of EmailOut. . . . . 45, 47  
 SI . . . . . 14  
 SourceContactId  
 attribute of MergeHistory . . . . . 29, 31  
 SrcContactContent  
 attribute of MergeHistory . . . . . 30, 31  
 SrcContactId  
 attribute of ContactArchiveSynchro . . . . 25  
 StandardResponseId  
 attribute of EmailOut. . . . . 46, 47  
 StartDate  
 attribute of Interaction . . . . . 34, 38  
 Status

attribute of Interaction. . . . . 34, 35  
 StoppedReason  
 attribute of Interaction. . . . . 35, 40  
 StrAttribute  
 attribute of Contact . . . . . 17, 19  
 attribute of Interaction. . . . . 35, 40  
 StructTextMimeType  
 attribute of Interaction. . . . . 34, 39  
 StructuredText  
 attribute of Interaction. . . . . 34, 39  
 StrValue  
 attribute of ContactAttribute. . . . . 22, 23  
 StrValueLowerCase  
 attribute of ContactAttribute. . . . . 22, 23  
 Subject  
 attribute of Interaction. . . . . 34, 38  
 Subtypeld  
 attribute of Interaction. . . . . 34, 36

**T**

TenantID  
 attribute of MergeHistory . . . . . 30  
 TenantId  
 attribute of Contact . . . . . 17, 18  
 attribute of ContactAttributeMetaData . . 19, 21  
 attribute of Interaction. . . . . 34, 39  
 attribute of MergeHistory . . . . . 29  
 Text  
 attribute of Interaction. . . . . 34, 38  
 TheComment  
 attribute of Interaction. . . . . 34, 39  
 TheName  
 attribute of ContactAttributeMetaData . . 19, 20  
 TheType  
 attribute of ContactAttributeMetaData . . 19, 20  
 ThreadHash  
 attribute of Interaction. . . . . 34, 39  
 ThreadId  
 attribute of Interaction. . . . . 34, 38  
 Timeshift  
 attribute of Interaction. . . . . 34, 38  
 ToAddresses  
 attribute of EmailIn . . . . . 43, 44  
 attribute of EmailOut . . . . . 45, 46  
 Typeld  
 attribute of Interaction. . . . . 34, 36  
 typographical styles . . . . . 9

**V**

VA . . . . . 14  
 Validated  
 attribute of MergeHistory . . . . . 30, 31  
 version numbering

document . . . . .	8
VMBT. . . . .	14

## **W**

WebSafeEmailStatus	
attribute of Interaction . . . . .	35, 40
WhichRuleMatched	
attribute of EmailIn . . . . .	43, 45