



## **BCC Architect**

# **BCC Architect Developer Guide**

United States

April – May 2020

COPYRIGHT ©2020 BCC Software, LLC  
75 Josons Drive  
Rochester, NY 14623-3494

This manual and software are copyrighted by BCC Software. All rights are reserved and neither manual nor software may be copied in any way without prior consent.

BCC Software and BCC Architect are registered trademarks of BCC Software, LLC. Track N Trace, BCC Software, and the BCC Software logo are trademarks of BCC Software in the United States and other countries.

TDbf software used under license. TDbf Copyright © 1991, 1999, Free Software Foundation, Inc. RapidJSON Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip. All rights reserved. Borland, dBase, and Paradox are registered trademarks of Borland International Incorporated. Microsoft, Windows, NT, MS, Access, Excel, and FoxPro are registered trademarks of Microsoft Corp. Zip and Jaz are trademarks of Iomega Corporation. Adobe® and Adobe PDF Library™ are trademarks or registered trademarks of Adobe Systems Inc. in the US and other countries. All printer brands or other product names mentioned herein may be trademarks or registered trademarks of their respective holders.

To the extent the software of BCC Software, LLC. described in this manual integrates data products and software of the United States Postal Service, such as RDI, DPV®, LACS<sup>Link</sup>®, RDI®, NCOA<sup>Link</sup> FSP®, NCOA<sup>Link</sup> LSP® with ANK<sup>Link</sup>®, DSF<sup>2</sup>®, eLOT®, Suite<sup>Link</sup>®, AIS Products, Labeling Lists, National Zone Charts Matrix Product, and AMS API®), you agree to be bound by the terms of the license agreements between BCC Software, LLC. and the United States Postal Service.

BCC Software is a non-exclusive licensee of the USPS for the following: NCOA<sup>Link</sup> Interface Developer and Distributor; NCOA<sup>Link</sup> Full and Limited Service Provider Licensee; LACS<sup>Link</sup>, DPV, and RDI™. DSF<sup>2</sup> services are provided by a non-exclusive licensee of the United States Postal Service and/or a direct license..

Prices for BCC Software products and services are not established, controlled or approved by the United States Postal Service or the United States Government.

For a list of trademarks owned by the United States Postal Service, please see Trademarks of the USPS: <https://postalpro.usps.com/mnt/glusterfs/2018-03/Trademarks.pdf> ↗.

The names, logos and international property rights of other companies regarding products and services remain the property of their respective owners.

202003020320

# Contents

Contents	ii
<b>Welcome to BCC Architect</b>	<b>1</b>
Installing BCC Architect	1
Product Updates	1
About this Guide	1
Additional Resources	1
<b>Processing Modes</b>	<b>2</b>
Processing Data Locally	2
Processing Data on an Internal Server	2
Processing Data on a BCC Software-Hosted Server	3
<b>Available Technologies</b>	<b>4</b>
COM	4
.NET	4
Web Services	5
TCP/IP	5
<b>Using BCC Architect for Common Tasks</b>	<b>7</b>
Initiating BCC Architect	7
Correcting Single Addresses in Real-Time	10
Correcting Batch Addresses	11
Updating Moved Addresses	13
Presorting Postal Mailings	15
Generating USPS Postal Reports	16
Reviewing Errors	16
Adding Geocode Data	17
Parsing or Building Phone Numbers	17
<b>COM API Reference</b>	<b>18</b>
The COM Object Factory	29
The COM ZipTask Object for Correcting Single Addresses	44
The COM CASSTask Object for Correcting Batch Addresses	155
The COM CASSReportTask Object for Creating USPS Postal Reports	203

The COM MOVETask Object for Updating Moved Addresses .....	222
The COM PresortTask Object for Presorting Mailings .....	257
The COM ReviewErrorsTask Object for Reviewing Errors .....	407
The COM GeocodeTask Object for Adding Geocode Data .....	417
The COM PhoneTask Object for Parsing or Building Phone Numbers .....	451
COM Field Names .....	463
COM MRTKTaskLib Result Codes .....	496
<b>.NET Classes Reference</b> .....	<b>507</b>
The .NET ZIPAssembly Class for Correcting Single Addresses .....	516
The .NET CASSAssembly Class for Correcting Batch Addresses .....	578
The .NET MOVEAssembly Class for Updating Moved Addresses .....	622
The .NET PRESORTAssembly Class for Presorting Mailings .....	668
<b>Web Services Reference</b> .....	<b>786</b>
The Web Services ZIPService Interface for Correcting Single Addresses .....	790
The Web Services CASSService Interface for Correcting Batch Addresses .....	819
The Web Services MOVEService Interface for Updating Moved Addresses .....	852
<b>Results Codes</b> .....	<b>903</b>
Address Correction Errors and Results .....	903
NCOALink Footnote Codes .....	909
NCOALink Move Type Codes .....	917
<b>Additional Resources</b> .....	<b>918</b>
Knowledge Base .....	918
How to Contact Support .....	918

# Welcome to BCC Architect

Welcome to BCC Architect for the United States.

This software offers large-scale integrations greater speed and flexibility for address cleansing and postal presorting solutions.

## Installing BCC Architect

BCC Architect is installed using an installer wizard. For help with installation, please see the Installation Guide.

## Product Updates

Every two months, you will receive a new issue of BCC Architect. To learn more about new features, refer to the Release Notes made available with the release of each issue.

## About this Guide

The Developer Guide is a comprehensive resource for developers using the BCC Architect API.

This manual describes the functions available through BCC Architect and serves as your primary API reference guide.

The first section of this guide contains a high-level overview of how BCC Architect works for some of the common tasks for which it is used.

The latter portion of this guide includes the complete API reference for COM, .NET, and Web Services.

## Additional Resources

The BCC Architect installer includes the Developer Center, which includes sample projects that demonstrate how to use each object with a variety of development technologies.

## Processing Modes

With BCC Architect, you can process data locally, via an internal server, or with services hosted by BCC Software. Depending on how you plan to process your data, you can select from several available technologies:

- [COM](#)
- [.NET](#)
- [Web Services](#)
- [TCP/IP](#)

### Processing Data Locally

Processing data in a single-machine local environment is often chosen for speed and security in an environment where one or a small number of computers need to access data source files. To process data locally, you can use several technologies available in BCC Architect, including:

- [COM](#)
- [.NET](#)

### Processing Data on an Internal Server

Processing data on your own server is often chosen for speed and security in an environment in which multiple computers need to access the same data source files. To process data locally, you can use several technologies available in BCC Architect, including:

- [COM](#)
- [.NET](#)
- [TCP/IP](#)

To use an internal server configuration, you will also need to implement the BCC Architect Server.

## BCC Architect Server

The BCC Architect Server is a Win32 application that currently accepts single address verification requests via a TCP/IP connection for the following countries: United States, Canada, Australia, and United Kingdom (with the purchase of these countries).

The clients of the server are very simple and can exist on any system. This simplicity allows you to have a Unix Web Server connect to the BCC Architect Server to process an address correction request.

The BCC Architect installer can install sample client code in the following programming languages:

- JAVA
- C/C++
- ASP

See the Sample Code section of the Developer Center. One of the sample projects is a Visual C++ application that will check addresses from any of the countries supported by your implementation of BCC Architect.

## BCC Architect Server Request Scheme Format

For a detailed description of the BCC Architect Server requests, please refer to the Client/Server Implementation Guide. The Implementation Guide is located in the Documents directory of the Developer Center.

## Processing Data on a BCC Software-Hosted Server

Processing data on a BCC Software-hosted server frees you from data maintenance. Using secure web services, you can use BCC Architect to process data using servers and data hosted by BCC Software.

The interface used for processing data via BCC Software-hosted servers is the Web Services interface.

BCC Software offers additional on-demand services. For more information about on-demand services, please contact BCC Software sales.

- [Web Services](#)

# Available Technologies

BCC Architect offers several technologies for processing data.

Please see [Processing Modes](#) for more information.

The technologies available include:

## COM

The BCC Architect COM interface supports all current Architect capabilities. COM is typically used in an environment where data will be processed locally or on an internal server.

- [Processing Data Locally](#)
- [Processing Data on an Internal Server](#)
- "Processing Data on a BCC Software-Hosted Server " on the previous page

For the BCC Architect COM reference, see:

- [COM API Reference](#)

For more information about COM, see:

- [The Microsoft® COM page](#)
- [The Microsoft Developer Network COM library](#)
- [The Wikipedia page on COM](#)

## .NET

The .NET assemblies in BCC Architect support most of the BCC Architect capabilities.

- [Processing Data Locally](#)
- [Processing Data on an Internal Server](#)

For the BCC Architect .NET API reference, see:

- [.NET Classes Reference](#)

For more information about .NET, see:

- [.NET Development](#)
- [The Wikipedia page on COM-related technologies](#)
- [The Wikipedia page about the .NET framework](#)
- [The Wikipedia page on .NET assembly](#)
- [The Stack Overflow Q&A about .NET assembly](#)

## Web Services

The Web Services in BCC Architect support some of the BCC Architect capabilities.

- ["Processing Data on a BCC Software-Hosted Server " on page 3](#)

For the BCC Architect Web Services API reference, see:

- [Web Services Reference](#)

BCC Software offers additional on-demand services. For more information, please contact [sales@bccsoftware.com](mailto:sales@bccsoftware.com).

For more information about Web Services, see:

- [The Wikipedia page on Web Services](#)
- [The Stack Overflow Q&A about Web Services](#)

## TCP/IP

TCP/IP is a server processing mode available for BCC Architect.

- [Processing Data on an Internal Server](#)

For more information about TCP/IP, see:

- [The Wikipedia page on TCP/IP](#)
- [The Wikipedia entry about the architectural principles of TCP/IP](#)

## Using BCC Architect for Common Tasks

BCC Architect has capabilities that allow you to cleanse address records, update moved addresses, presort mailings, and integrate with other environments. Depending on your licensing agreement for BCC Architect, you can use Architect for:

- [Correcting Single Addresses in Real-Time](#)
- [Correcting Batch Addresses Asynchronously](#)
- [Updating Moved Addresses](#)
- [Presorting Postal Mailings](#)
- [Creating USPS Postal Reports](#)
- [Reviewing Errors](#)
- [Adding Geocode Data](#)
- [Parsing or Building Phone Numbers](#)

## Initiating BCC Architect

Your registration keys are entered when you install BCC Architect. Whenever you use the BCC Architect API, the registration keys are checked automatically. If you try to create an object that is not part of your license agreement, you will receive an error notifying you that your registration key is invalid, expired, or missing.

## Validating Registration Keys

If you need to validate the registration information for your key:

1. Download the latest BCC Architect release (or insert the BCC Architect DVD into your disc drive). Click **Install** BCC Architect.
2. The Welcome screen displays. Select **Modify**, and click **Next** to continue.
3. The Registration Keys screen displays. This screen lists all of your registered keys and their expiration dates. Review the information you wish to validate.

- When you are done, click **Cancel**. Select **Yes**, and click **Finish** to exit the installation wizard.

## BCC Architect Add-Ons Extend Your Capabilities

BCC Architect allows extended functionality with add-ons. The following add-ons are available.

Add-On	Description
Mixed Weights add-on	<p>The Mixed Weight add-on allows you to sort mailings with variable piece weights and thicknesses.</p> <p>Mail pieces with differing weights and/or thickness may be charged different amounts of postage.</p>
Package Services add-on	<p>The Package Services add-on allows you to access additional mail sorts for packages. The mail sorts include Media Mail, Bound Printed Matter and Library Mail. These mail sorts are intended for larger mail pieces and offer postage rates lower than parcel rates in other mailing classes.</p>
Geocode add-on	<p>The Geocode add-on allows you to add longitude and latitude information to your address records.</p> <p>Geocoding adds the following fields to your address records:</p> <ul style="list-style-type: none"> <li>• Latitude</li> <li>• Longitude</li> <li>• Census Block Group</li> <li>• Census Tract</li> <li>• Metropolitan Statistical Area (MSA) number</li> </ul>

Add-On	Description
RDI add-on	<p>The RDI Interface add-on flags addresses as residential during Address Correction. Some mail carriers charge extra to deliver packages to residential addresses, so this information can help you determine the most cost effective option when shipping packages.</p> <p>Using RDI requires a licensing agreement; please see the Knowledge Base article about the simplified process, <a href="#">Taking Advantage of Simplified RDI</a> ↗, for more information.</p>
Palletization add-on	<p>The Palletization add-on allows you to further organize larger mailings into pallets. Pallets contain multiple mail containers sorted to the same level and can be easier to manage, especially if you have over 500 pounds of mail going to a specific location. You can receive additional discounts by using pallets.</p>
Mail.dat	<p>The Mail.dat add-on lets you upload electronic postal forms to the <i>PostalOne!</i> system. Using electronic documentation is a requirement for Full-Service Intelligent Mail.</p>
Track N Trace	<p>Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings.</p> <p>Two add-on keys are available:</p> <ul style="list-style-type: none"> <li>• Track N Trace</li> <li>• Track N Trace Reseller Site</li> </ul>

To learn more about add-on capabilities, contact BCC Software Sales.

### To Register BCC Architect Add-On Keys

If you have previously installed BCC Architect and want to add registration keys for add-ons, follow these steps.

1. Download the latest BCC Architect release (or insert the BCC Architect DVD into your disc drive). Click **Install** BCC Architect.
2. The **Welcome** screen displays. Select **Modify**, and click **Next** to continue.

3. The **Registration Keys** screen displays. Enter the registration keys for the add-ons you have registered, and click **Add** for each. When you have finished entering keys, click **Next** to continue.
4. Complete the installation wizard to register your keys.

If you are registering add-ons as part of a new installation, follow the normal installation procedure and enter your add-on keys where prompted.

## Correcting Single Addresses in Real-Time

You can use BCC Architect to correct individual addresses on a transactional basis, in real time. This address correction uses the BCC Software CASS™-Certified address correction engine.

The general process is to:

1. Create the object.
2. Set the address and properties.
3. Check the address.
4. Retrieve the corrected address record.

## Interfaces for Correcting Single Addresses

You can use any of the available interfaces to correct single addresses.

Please see the Developer Center for additional examples and sample code.

### .NET

1. Create a ZIPAssembly object.
2. Set the address and configuration properties.
3. Call CheckAddress to process the address.
4. Retrieve the corrected address from the ZIPAssembly object.

[See the .NET ZIPAssembly reference for more information about ZIPAssembly.](#)

## COM

1. Create a ZIPTask COM object.
2. Set the address and configuration properties.
3. Call CheckAddress to process the address.
4. Retrieve the corrected address from the ZIPTask object.

[See the COM ZIPTask reference for more information about ZIPTask.](#)

## Web Services

1. Create a ZIPService reference and a client-side web service interface object.
2. Set the address and configuration properties
3. Call CheckAddress to process the address.
4. Retrieve the corrected address from the web service.

[See the Web Services ZIPService reference for more information about ZIPService.](#)

## Correcting Batch Addresses

You can use BCC Architect to correct batches of addresses on a scheduled basis. This address correction uses the BCC Software CASS™-Certified address correction engine.

Batch address correction works by packaging multiple addresses and passing them along to the processing routines on the local machine or on a semi-dedicated server. After the processing of address correction is done, reports can be printed for use with a mailing.

The general process is to:

1. Create the object.
2. Set the address and properties.
3. Create objects for each address, and combine into an address block

4. Process the address block.
5. Retrieve the corrected address records.

## Interfaces for Correcting Batch Addresses

You can use any of the available interfaces to correct batch addresses.

Please see the Developer Center for additional examples and sample code.

### .NET

1. Create a CASSAssembly object.
2. Configure the address and configuration properties.
3. Create USAddressRecord objects for each address and combine into USAddressRecordBlock objects.
4. Process the USAddressRecordBlock object using the CASSAssembly object.
5. Retrieve your results using the resulting USAddressRecordBlock object and its contained USAddressRecord objects.

[See the .NET CASSAssembly reference for more information about CASSAssembly.](#)

### COM

1. Create a CASSTask COM object.
2. Configure the address and configuration properties.
3. Process the addresses.
4. Retrieve the corrected addresses from the COM object.
5. If you have selected them, view or print your reports.

[See the COM CASSTask reference for more information about CASSTask.](#)

## Web Services

1. Create a CASSService reference and a client-side web service interface object.
2. Configure the address and configuration properties.
3. Process the addresses.
4. Retrieve the corrected addresses from the Web Service.
5. If you have selected them, view or print your reports.

[See the WebServices CASSService reference for more information about CASSService.](#)

## Updating Moved Addresses

You can use BCC Architect to compare addresses against the National Change of Address (NCOA<sup>Link</sup>) service provided by USPS, and update moved addresses for up to the past 48 months.

Address updates are submitted to a server where they are recoded with updated address information.

To run Move Update, you must process at least 100 unique records.

The general process is to:

1. Create the object.
2. Set the move processing and configuration properties.
3. Create objects for each address, and combine into an address block.
4. Submit the blocks for MOVE processing.
5. Retrieve the updated address blocks.
6. Create or view reports, if needed.

## Interfaces for Updating Moved Addresses

You can use any of the available interfaces to update moved addresses.

Please see the Developer Center for additional examples and sample code.

## .NET

1. Create a MOVEAssembly object is created
2. Configure Move Update processing and general properties.
3. Create a USAddressRecord object d for each address that is to be process for Move Update. Add each address to a USAddressRecordBlock object.
4. Submit each USAddressRecordBlock for Move Update processing using the MOVEAssembly object.
5. Once the move update process is complete, retrieve USAddressRecordBlock objects from the MOVEAssembly object. Each contains a portion of the processed list.
6. You can then print or view reports.

[See the .NET MOVEAssembly reference for more information about MOVEAssembly.](#)

## COM

1. Create a MOVETask COM object.
2. Configure Move Update processing and general properties.
3. Process the addresses.
4. Retrieve the corrected addresses from the COM object.
5. If you have selected them, view or print your reports.

[See the COM MOVETask reference for more information about MOVETask.](#)

## Web Services

1. Create a Move Service referenceand a client-side web service interface object.
2. Configure Move Update processing and general properties.
3. Process the addresses.

4. Retrieve the corrected addresses from the Web Service.
5. If you have selected them, view or print your reports.

[See the Web Services MOVEService reference for more information about MOVEService.](#)

## Presorting Postal Mailings

You can use BCC Architect to sort and prepare your address records for a USPS postal mailing.

The general process is to:

1. Create the object.
2. Set the sorting and configuration properties.
3. Create objects for each address, and combine into an address block.
4. Submit the blocks for PRESORT processing.
5. Retrieve the sorted address block.
6. Create or view reports, if needed.

## Interfaces for Presorting Postal Mailings

You can use .NET or COM interfaces to presort postal mailings.

Please see the Developer Center for additional examples and sample code.

### .NET

1. Create a PRESORTAssembly object.
2. Configure the sorting and general properties.
3. Create USAddressRecord objects for each address in the sort and combine into USAddressRecordBlock objects.
4. Submit each USAddressRecordBlock for sorting using the PRESORTAssembly object.
5. Retrieve the USAddressRecordBlock objects from the PRESORTAssembly object after

sorting is complete. Each USAddressRecordBlock contains a portion of the sort results.

6. If you have selected them, view or print your reports.

[See the .NET PRESORTAssembly reference for more information about PRESORTAssembly.](#)

## COM

1. Create a PresortTask COM object.
2. Set the address and configuration properties.
3. Process the object.
4. Retrieve the corrected and sorted addresses from the COM object.
5. A PresortTask COM object is created
6. If you have selected them, view or print your reports.

[See the COM PresortTask reference for more information about PresortTask.](#)

## Generating USPS Postal Reports

When you have completed a postal presort, you can use BCC Architect to generate USPS postal reports.

### Interfaces for Creating USPS Postal Reports

You can use COM interfaces to generate USPS postal reports.

[See the COM ReportTask reference for more information about ReportTask.](#)

## Reviewing Errors

The ReviewErrorsTask object in BCC Architect enables you to create scripts for viewing and correcting uncorrected or questionable addresses .

### Interfaces for Reviewing Errors

You can use the COM interfaces to review errors.

[See the COM ReviewErrors reference for more information about ReviewErrors.](#)

## Adding Geocode Data

The BCC Architect Geocode object returns the latitude and longitude coordinates and related data for a given ZIP Code. It can also compute the distance or angle between two points and apply radius-based filters. Geocode is a BCC Architect add-on.

### Interfaces for Adding Geocode Data

You can use COM interfaces to add Geocode data.

[See the COM GeocodeTask reference for more information about GeocodeTask.](#)

## Parsing or Building Phone Numbers

The BCC Architect PhoneTask object can parse a phone number into its individual elements. It can also build a phone number from its individual elements, using several different formatting options.

### Interfaces for Parsing or Building Phone Numbers

You can use COM interfaces to parse or build phone numbers.

[See the COM PhoneTask reference for more information about PhoneTask.](#)

# COM API Reference

## Contents

The COM Object Factory .....	29
Configuring Object Factory Tasks .....	29
Object Factory Functions .....	30
CreateObject .....	30
GetAddOnState .....	31
GetDaysRemaining .....	32
IsObjectExpired .....	33
IsUpdateAvailable .....	34
RunUpdate .....	34
Object Factory Properties .....	35
DataFilePath .....	35
FileSearchPath .....	37
MRTKRegistrationKey .....	37
Object Factory Object Names Summary Table .....	38
Object Factory tagMRTKOBJECTS Properties Summary Table .....	40
Object Factory Country IDs Summary Table .....	42
Object Factory Global Properties Summary Table .....	42
The COM ZipTask Object for Correcting Single Addresses .....	44
ZIPTask Sample ASP Code .....	45
ZIPTask Functions .....	46
PrepareTask .....	46
BrowseAddress .....	47
BuildAddress .....	48
CheckAddress .....	49
ClearAddress .....	51
ParseAddress .....	52
EndTask .....	54
ZIPTask Properties .....	54
AddressBlock .....	54
AddressInputPreference .....	58
AddressLine1 .....	59
AddressLine2 .....	61
AddressLineAbbreviated .....	62
AddressSuggestionList .....	64
ApplyCasingBusiness .....	65
AssignCounty .....	66
AssignLOT .....	67
AssignRDI .....	68
BusinessName .....	69
Capitalize .....	70
CarrierRoute .....	71
CarrtCoded .....	72
Casing .....	73
CASSDate .....	74
CensusBlock .....	75
CensusTract .....	76
CertifyFlag .....	77
City .....	79

CityFormat	80
CityStateZip	81
CongressionalDistrict	82
CountyCode	83
CountyFIPSCode	84
CountyName	85
DatafileLocation	85
DeliveryPointCheckDigit	86
DPBarcodeString	87
DPVCoded	88
DPVFailureAsError	89
DPVFootnotes	90
DPVIndicator	91
DPVIsCMRA	93
DPVIsDoorNotAccessible	94
DPVIsNoSecureLocation	95
DPVIsNoStat	96
DPVIsThrowback	97
DPVIsVacant	98
DPVLocation	99
DPVResolveMultipleResponse	100
ErrorCodes	101
ErrorCodeString	101
ExtralInfo	102
FirmOutput	103
GeocodeFootnote	104
HighwayContractFormat	106
IsResidence	107
KeepAliasAddress	108
KeepExtraPrimaryData	109
KeepNonMailingCity	110
LACS	111
LACSFootnote	112
LACSIndicator	113
Latitude	114
Longitude	115
LOTNumber	117
MSACode	118
MailRoomServer	119
MatchedToDefault	120
Plus4Coded	121
PMB	122
PMBOutput	122
POBoxDeliveryOnlyZip	124
POBoxFormat	124
PostDirectional	125
PostDirectionalFormat	126
PreDirectional	127
PreDirectionalFormat	128
PrimaryAddressOutput	129
PrimaryNumber	131
RecordType	132
ReturnAddressSuggestionList	133
ReturnInputAddressOnUnconfirmedDPV	134

RuralRouteFormat .....	135
SilentMode .....	136
State .....	137
Street .....	138
Suffix .....	139
SuffixFormat .....	140
SuiteLinkFootnote .....	141
UnitDesignator .....	142
UnitDesignatorFormat .....	143
UnitNumber .....	144
UnitOutput .....	145
UpdateUncorrectedCityStZip .....	147
Urbanization .....	148
UseDPV .....	149
UseGeocode .....	150
UseLACS .....	151
UseSuiteLink .....	152
ZipCode .....	153
ZipCoded .....	154
<b>The COM CASSTask Object for Correcting Batch Addresses</b> .....	<b>155</b>
CASSTask Functions .....	155
PrepareTask .....	156
GetProperty .....	156
SetProperty .....	157
ShowCASSWizard .....	158
ValidateProperties .....	158
GetPropertySummary .....	159
Update .....	160
UpdateV .....	162
ReviewErrors .....	163
RetrieveReviewed .....	164
RetrieveReviewedV .....	165
Print3553 .....	167
PrintPreview3553 .....	168
SaveReportAsPDF .....	168
SaveReportsAsPDF .....	170
AbortTask .....	171
EndTask .....	172
CASSTask Properties .....	172
ctABBREVIATE_ADDRESS_LINE .....	172
ctADDRESS_INPUT_PREFERENCE .....	173
ctALL_CAPS .....	173
ctCASING .....	174
ctCASS_WIZARD_CAPTION .....	174
ctCERTIFY_FLAG .....	175
ctCITY_FORMAT .....	175
ctDPV_FAILURE_AS_ERROR .....	176
ctDPV_RESOLVE_MULTIPLE_RESPONSE .....	176
ctFIRM_OUTPUT .....	177
ctHIDE_PROGRESS_AFTER_BATCH .....	177
ctHIGHWAYCONTRACT_FORMAT .....	177
ctKEEP_ALIAS_ADDRESS .....	178
ctKEEP_EXTRA_PRIMARY_DATA .....	178
ctLIST_NAME .....	179

ctLIST_PROCESSOR .....	179
ctMAILERS_ADDRESS .....	179
ctMAILERS_CITY .....	180
ctMAILERS_NAME .....	180
ctMAILERS_STATE .....	180
ctMAILERS_ZIP .....	181
ctMRTK_VERSION .....	181
ctPMB_OUTPUT .....	182
ctPOBOX_FORMAT .....	182
ctPOSTDIRECTIONAL_FORMAT .....	182
ctPREDIRECTIONAL_FORMAT .....	183
ctPRIMARY_ADDRESS_LENGTH_LIMIT .....	183
ctPRIMARY_ADDRESS_OUTPUT .....	184
ctRECORD_COUNT .....	184
ctRETURN_INPUT_ADDRESS_ON_UNCONFIRMED_DPV .....	185
ctREVIEW_ERRORS .....	185
ctRURALROUTE_FORMAT .....	185
ctSHOW_PROGRESS .....	186
ctSILENT_MODE .....	186
ctSUFFIX_FORMAT .....	186
ctUNIT_DESIGNATOR_FORMAT .....	187
ctUNIT_OUTPUT .....	187
ctUPDATE_UNCORRECTED_CITYSTZIP .....	188
ctUPDATECASE_BUSINESS .....	188
ctUPDATECASE_NAMES .....	188
ctUSE_COUNTRY .....	189
ctUSE_SUITELINK .....	189
COM CASSTask Properties Summary Table .....	190
<b>The COM CASSReportTask Object for Creating USPS Postal Reports</b> .....	<b>203</b>
CASSReportTask Functions .....	203
PrepareTask .....	203
GetProperty .....	204
SetProperty .....	205
ValidateProperties .....	206
Print3553 .....	206
PrintPreview3553 .....	207
EndTask .....	208
CASSReportTask Properties .....	208
crLIST_NAME .....	208
crLIST_PROCESSOR .....	209
crMAILERS_ADDRESS .....	209
crMAILERS_CITY .....	209
crMAILERS_NAME .....	210
crMAILERS_STATE .....	210
crMAILERS_ZIP .....	210
crPRINTER_NAME .....	211
crSHOW_PAGE_SETUP .....	211
crSHOW_PRINT_DIALOG .....	211
COM MRTK Report IDs Summary Table .....	212
COM CASSReportTask Properties Summary Table .....	214
COM ReportWizard Properties Summary Table .....	214
<b>The COM MOVETask Object for Updating Moved Addresses</b> .....	<b>222</b>
MOVETask Functions .....	223
PrepareTask .....	223

GetProperty .....	224
SetProperty .....	224
ShowMOVEWizard .....	225
ValidateProperties .....	226
Send .....	227
DoProcess .....	228
DoProcessEx .....	229
Retrieve .....	230
Print3553 .....	231
PrintActivationReport .....	232
PrintNDIForm .....	233
PrintPreview3553 .....	234
PrintPreviewActivationReport .....	234
PrintPreviewNDIForm .....	235
SaveReportAsPDF .....	236
PrintPreviewReports .....	238
PrintReports .....	238
SaveReportsAsPDF .....	239
EndTask .....	241
MOVETask Properties .....	241
nIMRTK_VERSION .....	241
nINCOA_BUYER_NAME .....	242
nINCOA_CLIENT_ID_LIST .....	242
nINCOA_CLIENT_ID_NAME_LIST .....	242
nINCOA_CUSTOMER_ADDRESS .....	243
nINCOA_CUSTOMER_EMAIL .....	243
nINCOA_CUSTOMER_FAX .....	243
nINCOA_CUSTOMER_LASTLINE .....	244
nINCOA_CUSTOMER_MAILERID .....	244
nINCOA_CUSTOMER_NAME .....	244
nINCOA_CUSTOMER_PHONE .....	245
nINCOA_CUSTOMER_SIC .....	245
nINCOA_CUSTOMER_TITLE .....	245
nINCOA_HIGH_MATCH_RATE_REASON .....	245
nINCOA_KEEP_ALTERNATE_ADDRESS_LINE .....	246
nINCOA_MAILCLASS .....	246
nINCOA_MOVE_MONTH_RANGE .....	247
nINCOA_MULTI_NAME_HANDLE .....	248
nINCOALINK_ADMIN_ID .....	248
nINCOALINK_ADMIN_PASSWORD .....	249
nINCOALINK_BROKER_ID .....	249
nINCOALINK_BROKER_PASSWORD .....	249
nINCOALINK_CUSTOMER_ID .....	250
nINCOALINK_CUSTOMER_PASSWORD .....	250
nINCOALINK_HIDE_PROGRESS_AFTER_PROCESS .....	250
nINCOALINK_MATCH_FLAG .....	251
nINCOALINK_SHOW_PROGRESS .....	251
COM MOVETask Properties Summary Table .....	252
The COM PresortTask Object for Presorting Mailings .....	257
PresortTask Functions .....	258
PrepareTask .....	258
GetProperty .....	258
SetProperty .....	259
ValidateProperties .....	260

Send .....	261
SendV .....	262
DoSort .....	263
Retrieve .....	264
RetrieveV .....	265
ShowPresortWizard .....	266
ShowReportWizard .....	267
ShowLabelWizard .....	268
GetPropertySummary .....	268
PrintReport .....	269
PreviewReport .....	270
SaveReportAsPDF .....	271
PrintPresortReports .....	273
PreviewPresortReports .....	274
SaveReportsAsPDF .....	275
AbortTask .....	277
EndTask .....	278
PresortTask Properties .....	278
ptACS_METHOD .....	278
ptALWAYS_USE_PERIODICALS_FSS_PREP .....	280
ptANNUAL_NONSUBSCRIBER_THRESHOLD_EXCEEDED .....	280
ptAPPLY_BREAK_MARK_INDICATOR .....	281
ptAPPLY_PARCEL_SURCHARGE .....	281
ptBREAK_MARK_IND_BUNDLE .....	282
ptBREAK_MARK_IND_CONTAINER .....	282
ptBREAK_MARK_IND_PALLET .....	282
ptCASS_CERTIFY_FIRST .....	283
ptCASS_PROCESS_DATE_AUTOMATION .....	283
ptCASS_PROCESS_DATE_ECR .....	283
ptCOMBINE_RESIDUAL_PIECES .....	284
ptCONFIRM_TRACKING .....	284
ptCONTAINER_LABEL_LAYOUT .....	285
ptCUSTOM_BARCODE_MAILER_ID .....	285
ptCREATE_COURTESY_PALLETS .....	286
ptCUSTOM_BARCODE_MAILER_ID .....	286
ptDATA_SERVICES_CLIENT_ID_LIST .....	286
ptDATA_SERVICES_CLIENT_USER .....	287
ptDATA_SERVICES_JOB_ID .....	287
ptDATA_SERVICES_PASSWORD .....	287
ptDATA_SERVICES_USER .....	288
ptDrop_SHIP_ADC_ZIPS .....	288
ptDROP_SHIP_BMC_ZIPS .....	288
ptDROP_SHIP_NDC_ZIPS .....	289
ptDROP_SHIP_SCF_ZIPS .....	289
ptENABLE_CASS_PROCESSING .....	289
ptENABLE_TEMPLATES .....	290
ptEXCEPTIONAL_DISPATCH_ZIPS .....	290
ptFIRM_BUNDLE_MIN .....	290
ptFORCE_WALK_SEQUENCE_SATURATION .....	291
ptFORM_NAME .....	291
ptHIDE_SORT_PROGRESS_AFTER_SORT .....	291
ptIM_BARCODE_MAILER_ID_CODE .....	292
ptIM_CONTAINER_SEQUENCE_LAST .....	292
ptIM_CONTAINER_SEQUENCE_START .....	293

ptIM_EINDUCTION .....	293
ptIM_EINDUCTION_ACCEPT_MISSHIPPED .....	294
ptIM_EINDUCTION_FAST_SCHEDULER_ID .....	294
ptIM_PALLET_SEQUENCE_LAST .....	294
ptIM_PALLET_SEQUENCE_START .....	295
ptIM_PIECE_SEQUENCE_LAST .....	295
ptIM_PIECE_SEQUENCE_START .....	295
ptIM_SEQUENCING_METHOD .....	296
ptINCOUNTY_ZIPS .....	296
ptJOB_ID .....	297
ptLABELS_BARCODE .....	297
ptLABELS_ENDORSEMENT .....	297
ptLEGACY_DDU_SUPPORT .....	298
ptMAIL_CONTENT .....	298
ptMAIL_OWNER_CRID .....	299
ptMAIL_OWNER_MAILER_ID .....	299
ptMAILDAT_CONTACT_EMAIL .....	300
ptMAILDAT_CONTACT_NAME .....	300
ptMAILDAT_CONTACT_PHONE .....	300
ptMAILDAT_CREATE_PBC .....	300
ptMAILDAT_CREATE_PDR .....	301
ptMAILDAT_INFORMED_CODE .....	301
ptMAILDAT_INFORMED_END .....	301
ptMAILDAT_INFORMED_ID .....	302
ptMAILDAT_INFORMED_NAME .....	302
ptMAILDAT_INFORMED_REPRESENT .....	303
ptMAILDAT_INFORMED_RIDEALONG .....	303
ptMAILDAT_INFORMED_START .....	304
ptMAILDAT_INFORMED_TARGETURL .....	304
ptMAILDAT_INFORMED_TITLE .....	304
ptMAILDAT_MAILING_FACILITY_ID .....	305
ptMAILDAT_MAILING_TITLE .....	305
ptMAILDAT_PERMIT HOLDER_ID .....	305
ptMAILDAT_USER_LICENSE_CODE .....	306
ptMAILDAT_VERIFICATION_FACILITY_NAME .....	306
ptMAILDAT_VERIFICATION_FACILITY_ZIP4 .....	306
ptMAILDAT_VERSION .....	307
ptMAILER_ID_USED .....	307
ptMAILING_AGENT_CRID .....	307
ptMAILING_AGENT_MAILER_ID .....	308
ptMAILING_IS_PLUS_ONE_MARRIAGE_MAIL .....	308
ptMANIFEST_SEQUENCE_NUMBER .....	308
ptMOVE_UPDATE_DATE .....	309
ptMOVE_UPDATE_METHOD .....	309
ptMRTK_PREFER_CONDENSED_REPORTS .....	310
ptMRTK_VERSION .....	310
ptOWNER_GHOST_NUMBER .....	311
ptPALLET_PLACARD_LAYOUT .....	311
ptPAUSE_BEFORE_FACING_SLIPS .....	311
ptPAUSE_BEFORE_PALLET_LABELS .....	312
ptPAUSE_BEFORE_TRAY_LABELS .....	312
ptPERMIT_SECONDARY_CITY .....	313
ptPERMIT_SECONDARY_ACCOUNT_NUMBER .....	313
ptPERMIT_SECONDARY_NUMBER .....	313

ptPERMIT_SECONDARY_STATE .....	314
ptPERMIT_SECONDARY_ZIP .....	314
ptPOSTALONE_JOB_ID .....	314
ptPERMIT_ZIPCODE .....	315
ptPREFERRED_CONTAINER_TYPE .....	315
ptPRESORT_SEQUENCING_DATE_ECR .....	316
ptPRESORT_TEMPLATE_LIST .....	316
ptPRINT_ALL_REPORTS .....	316
ptPRINT_CASS .....	317
ptPRINT_FACING_SLIPS .....	317
ptPRINT_FILE_TRAYLABELS .....	317
ptPRINT_MANIFEST .....	318
ptPRINT_PALLET_LABELS .....	318
ptPRINT_POSTAGE .....	318
ptPRINT_POSTAGE_SUMMARY .....	319
ptPRINT_PRESORT .....	319
ptPRINT_QUALIFICATION .....	319
ptPRINT_SETTINGREPORT .....	320
ptPRINT_TRAY_LABELS_ONLY .....	320
ptPRINT_TRAYLABELS .....	320
ptPRINT_ZONE .....	321
ptPUB_CONTACT_NAME .....	321
ptPUB_CONTACT_PHONE .....	321
ptPUB_EDITION_ISSUE .....	322
ptPUB_ENTRY_STATE_ZIP4 .....	322
ptPUB_ISSUE_DATE .....	322
ptPUB_ISSUE_FREQUENCY .....	323
ptPUB_OWNER_AGENT_NAME .....	323
ptPUB_TITLE .....	323
ptPUB_VOLUME_NUMBER .....	324
ptRECORD_COUNT .....	324
ptRECORD_COUNT_PER_RECEIVE .....	324
ptREPORT_FILE_TRAYLABELS .....	325
ptREPORT_FILE_NAME_MAILDAT .....	325
ptREPORT_FOLDER_NAME_MAILDAT .....	325
ptREPORT_MAILDAT_SAVE_AS_ZIP .....	326
ptREPORT_PRINTER_CASS .....	326
ptREPORT_PRINTER_DEFAULT .....	326
ptREPORT_PRINTER_MANIFEST .....	327
ptREPORT_PRINTER_POSTAGE .....	327
ptREPORT_PRINTER_PRESORT .....	327
ptREPORT_PRINTER_QUALIFICATION .....	328
ptREPORT_PRINTER_SETTINGREPORT .....	328
ptREPORT_PRINTER_TRAYLABELS .....	328
ptREPORT_PRINTER_ZONE .....	329
ptREPORT_SAVE_MAILDAT .....	329
ptREPORT_WIZARD_CAPTION .....	329
ptRIDEALONG_WEIGHT .....	330
ptSHOW_PAGE_SETUP .....	330
ptSAVE_SHIPPING_SERVICES_FILE .....	330
ptSHIPPING_SERVICES_FILE_PATH .....	331
ptSHIPPING_SERVICES_LOGIN_ID .....	331
ptSHIPPING_SERVICES_SEQUENCE_NUMBER .....	331
ptSHOW_PRINT_DIALOG .....	332

ptSHOW_SORT_PROGRESS .....	332
ptSILENT_MODE .....	333
ptSIMPLIFIED_DELIVERY_STATS_DATE .....	333
ptSORT_RESULTS_FLAG .....	333
ptSORT_WIZARD_CAPTION .....	334
ptSTATEMENT_NUMBER .....	334
ptSUPPRESS_CARRT_BASIC_RATES .....	334
ptSUPPRESS_INCOUNTY_RATES .....	335
ptTEMPLATE_NAME_TO_USE .....	335
ptUSE_DETACHED_ADDRESS_LABEL .....	336
ptUSE_EASYTRACK .....	336
ptUSE_EMM_TRAYS .....	337
ptUSE_EXCEPTIONAL_DISPATCH .....	338
ptUSE_FIRM_BUNDLES .....	338
ptUSE_MAILDAT .....	338
ptUSE_REPOSITIONABLE_NOTES .....	339
ptUSE_SIMPLIFIED_ADDRESSING_PIECE_MAX .....	339
ptUSE_USPS_PROMOTION .....	340
Presort Wizard Properties .....	340
pwBUNDLE_PIECE_MAX .....	340
pwCARTON_PIECE_MAX .....	341
pwCARTON_WEIGHT_MAX .....	341
pwCREATE_CREATE_FSF_SACKS .....	341
pwCREATE_LOW_VOLUME_PALLETS .....	342
pwCREATE_ORIGIN_DESTINATION_CONTAINERS .....	342
pwPRESORT_ADC_ZIP .....	342
pwPRESORT_BMC_ZIP .....	343
pwPRESORT_CLASS .....	343
pwPRESORT_COUNTY_NAME .....	343
pwPRESORT_DDU_ZIP .....	343
pwPRESORT_DEFAULT_MIN_WEIGHT .....	344
pwPRESORT_ENTRY_POINT_BMC .....	344
pwPRESORT_ENTRY_POINT_NDC .....	344
pwPRESORT_ENTRY_ZIP_CODE .....	345
pwPRESORT_INTELLIGENT_MAIL_ONLY .....	345
pwPRESORT_MAKE_ONLY_FULL_5_DIGIT_CARRIER_ROUTE_TRAYS .....	345
pwPRESORT_MAX_PER_TRAY .....	346
pwPRESORT_MIN_PER_TRAY .....	346
pwPRESORT_MIXED_WEIGHT_SORT .....	346
pwPRESORT_MULTIPLE_ENTRY_POINT .....	346
pwPRESORT_NDC_ZIP .....	347
pwPRESORT_NONMACHINABLE .....	347
pwPRESORT_OPTION .....	347
pwPRESORT_ORIGIN_KEY .....	348
pwPRESORT_PIECES_ARE_BARCODED .....	348
pwPRESORT_PIECES_INCH .....	348
pwPRESORT_PIECE_THICKNESS .....	349
pwPRESORT_RATE .....	349
pwPRESORT_REDUCE_OVERFLOW_CONTAINERS .....	349
pwPRESORT_SCF_ZIP .....	350
pwPRESORT_TYPE .....	350
pwPRESORT_WEIGHT_PER_PIECE .....	350
pwPRESORT_WS_SORT .....	350
pwSUMMARY_PIECE .....	351

pwSUMMARY_REPORTS .....	351
pwSUMMARY_SORT .....	351
Report Wizard Properties .....	352
rwAGENT_ADDRESS_ADDR .....	352
rwAGENT_ADDRESS_CITY .....	352
rwAGENT_ADDRESS_CONTACT .....	352
rwAGENT_ADDRESS_EMAIL .....	353
rwAGENT_ADDRESS_NAME .....	353
rwAGENT_ADDRESS_PHONE .....	354
rwAGENT_ADDRESS_STATE .....	354
rwAGENT_ADDRESS_ZIP .....	354
rwMAILING_DATE .....	355
rwNEWSPAPER_LABELS .....	355
rwNON_ADVERTISE_AMOUNT .....	355
rwORG_ADDRESS_ADDR .....	355
rwORG_ADDRESS_CITY .....	356
rwORG_ADDRESS_CONTACT .....	356
rwORG_ADDRESS_EMAIL .....	357
rwORG_ADDRESS_NAME .....	357
rwORG_ADDRESS_PHONE .....	357
rwORG_ADDRESS_STATE .....	358
rwORG_ADDRESS_ZIP .....	358
rwORG_MAILER_ID .....	358
rwORG_NONPROFIT_AUTH_NO .....	359
rwPAYMENT_ACCOUNT_NUMBER .....	359
rwPAYMENT_OPTION .....	360
rwPAYMENT_TYPE .....	360
rwPERMIT_ADDRESS_ADDR .....	361
rwPERMIT_ADDRESS_CITY .....	361
rwPERMIT_ADDRESS_CONTACT .....	361
rwPERMIT_ADDRESS_EMAIL .....	362
rwPERMIT_ADDRESS_NAME .....	362
rwPERMIT_ADDRESS_STATE .....	362
rwPERMIT_ADDRESS_ZIP .....	363
rwPERMIT_CAPS_CUSTOMER_ID .....	363
rwPERMIT_MAILER_ID .....	363
rwPERMIT_NONPROFIT_AUTH_NO .....	364
rwPERMIT_NUMBER .....	364
rwPOST_OFFICE .....	364
rwPOSTAGE_AMOUNT .....	365
rwPRINT_PRESORT_CITYSTATE .....	365
rwPRINT_PRESORT_MAILER_ID .....	365
rwPRINT_PRESORT_NAME .....	366
rwPRINT_PRESORT_PUB_ID .....	366
rwPRINT_PRESORT_ZIPCODE .....	366
rwSACK_LABEL_COLUMNS .....	367
rwSACK_LABEL_CONTINUOUS .....	367
rwSACK_LABEL_HEIGHT .....	367
rwSACK_LABEL_LEFT_MARGIN .....	368
rwSACK_LABEL_ROWS .....	368
rwSACK_LABEL_TOP_MARGIN .....	368
rwSACK_LABEL_WIDTH .....	369
rwTELEPHONE .....	369
rwTRAY_LABEL_COLUMNS .....	369

rwTRAY_LABEL_CONTINUOUS .....	370
rwTRAY_LABEL_HEIGHT .....	370
rwTRAY_LABEL_LEFT_MARGIN .....	370
rwTRAY_LABEL_ROWS .....	371
rwTRAY_LABEL_TOP_MARGIN .....	371
rwTRAY_LABEL_WIDTH .....	371
COM PresortTask Properties Summary Table .....	372
COM PresortWizard Properties Summary Table .....	403
The COM ReviewErrorsTask Object for Reviewing Errors .....	407
ReviewErrorsTask Functions .....	407
PrepareTask .....	407
GetProperty .....	408
SetProperty .....	408
ValidateProperties .....	409
Send .....	410
SendV .....	411
ReviewErrors .....	412
Retrieve .....	413
RetrieveV .....	414
EndTask .....	415
ReviewErrorsTask Properties .....	415
etRECORD_COUNT_PER_RECEIVE .....	416
etSHOW_RECEIVE_PROGRESS .....	416
COM ReviewErrorsTask Properties Summary Table .....	416
The COM GeocodeTask Object for Adding Geocode Data .....	417
GeocodeTask Functions .....	417
PrepareTask .....	417
AddIsBetweenFilter .....	417
AddRadiusExcludeFilter .....	419
AddRadiusIncludeFilter .....	420
AddRadiusIntersectFilter .....	421
BearingDegreesFromBaseToPoint .....	423
BearingDegreesFromBaseToZipCode .....	424
BearingDirectionFromBaseToPoint .....	425
BearingDirectionFromBaseToZipCode .....	427
CheckGeocode .....	429
ClearFilters .....	430
ClearGeocode .....	431
ConvertDoubleToString .....	432
ConvertLongToString .....	433
ConvertMilesToKilometers .....	433
ConvertStringToDouble .....	434
ConvertStringToLong .....	435
DistanceFromBaseToPoint .....	435
DistanceFromBaseToZipCode .....	436
IsPointWithinFilters .....	437
IsZipCodeWithinFilters .....	439
EndTask .....	440
GeocodeTask Properties .....	440
CountyFIPSCode .....	441
Footnotes .....	442
Latitude .....	443
Longitude .....	444
MSACode .....	445

RecordType .....	446
CensusBlock .....	447
CensusTract .....	448
ZipCode .....	449
The COM PhoneTask Object for Parsing or Building Phone Numbers .....	451
PhoneTask Functions .....	451
PrepareTask .....	451
ClearPhone .....	451
FormatPhone .....	452
ParsePhone .....	453
EndTask .....	454
PhoneTask Properties .....	455
AreaCode .....	455
ErrorCode .....	456
Extension .....	457
Format .....	458
LineNumber .....	460
Number .....	461
Prefix .....	462
COM Field Names .....	463
COM MRTKTaskLib Result Codes .....	496

## The COM Object Factory

The MRTKObjFactory object creates all BCC Architect Task objects. The MRTKObjFactory properly initializes interface objects and runs AutoUpdate on clients. The AutoUpdate feature automatically updates clients to make sure they have the most current Task interfaces and data files.

The BCC Architect object factory is contained in the MRTKFact.dll.

## Configuring Object Factory Tasks

You need to follow these basic steps to configure each Task object:

1. Create an object through the MRTK Object Factory.
2. Once an object has been created, call its PrepareTask function.
3. Now set the Task object's properties through calls to SetProperty. See the tables at the end of this document for descriptions of the available properties for each Task object.
4. After setting the relevant properties for a Task, call ValidateProperties.
5. Perform the work of the particular task. If you were using the CASSTask, for instance, you would call Update to CASS process an address block. For more information on what this requires, see the section on that particular task.

6. Call `EndTask` once you have finished using a `Task` object.
7. For a detailed description of the purpose and configuration of each `Task` object, we recommend that you read the introduction to each object.

## Object Factory Functions

The `MRTKObjFactory` functions are defined below. Once you have added a reference to the `MRTKFact.dll`, you can begin using these functions in your project.

### CreateObject

#### Syntax

```
CreateObject (eMRTKObject)  
  
IDispatch CreateObject (tagMRTKOBJECTS eMRTKObject)
```

#### Description

Creates a `Task` object.

#### Parameters

*eMRTKObject as tagMRTKOBJECTS*

`eMRTKObject` is the type of object to create.

*eMRTKObject*

`eMRTKObject` is the type of object to create.

#### Return values

Newly created interface of type `eMRTKObject`, otherwise nothing.

#### Notes

You must create all interfaces using this method for a few reasons. The first and most important is that the `CreateObject` function is responsible for running the `AutoUpdate` feature of `BCC Architect`. If `CreateObject` is not used to initialize an interface, then the `AutoUpdate` feature must be implemented by calling `IsUpdateAvailable`, and if `IsUpdateAvailable` returns `True`, then calling `RunUpdate`.

#### See also

See the [Object Factory tagMRTKOBJECTS Properties Summary Table](#) for a list of possible objects.

## GetAddOnState

### Syntax

```
GetAddOnState (nAddOnID)

long GetAddOnState (tagMRTKADDONS nAddOnID, long *nAddOnState)
```

### Description

Checks the status of an add-on.

### Parameters

*nAddOnID as tagMRTKADDONS*

The add-on to check, as specified by the table below.

*nAddOnID*

The add-on to check, as specified by the table below.

*nAddOnState*

Returns the status of the add-on:

Value	Description
-1	Add-on is not installed
0	Add-on is installed but has expired
>0	Days remaining until expiration of add-on

### Return values

Returns the state of the add-on (as Long):

Value	Description
-1	Add-on is not installed
0	Add-on is installed but has expired
>0	Days remaining until expiration of add-on

The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

Specify the BCC Architect add-on using the tagMRTKADDONS enumeration as defined in the following table:

Name	Value	Description
ADDON_MIXED_WEIGHTS	63	Mixed Weights add-on
ADDON_PACKAGE_SVCS	64	Package Services add-on
ADDON_GEOCODE	66	Geocode add-on
ADDON_RDI	68	RDI add-on
ADDON_PALLETIZATION	70	Palletization add-on

## GetDaysRemaining

### Syntax

```
GetDaysRemaining(nDaysRemaining)  
int GetDaysRemaining(long *nDaysRemaining)
```

### Description

Retrieves the number of days before the data files expire.

### Parameters

*nDaysRemaining* as int

Returns the number of days until the data files expire.

*nDaysRemaining*

Returns the number of days until the data files expire.

### Return values

None. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

Current US Postal regulations stipulate that CASS data files expire every two (2) months. This function can be used to take specific action when new data files are required.

## See also

[IsObjectExpired](#)

## IsObjectExpired

### Syntax

```
IsObjectExpired (eMRTKObject)  
long IsObjectExpired (tagMRTKOBJECTS eMRTKObject)
```

### Description

Checks if the specified BCC Architect object has expired.

### Parameters

*eMRTKObject as TagMRTKOBJECTS*

The type of object to check for expiration.

*eMRTKObject*

The type of object to check for expiration.

### Return values

0 if object has not expired, 1 if object has expired (as Long). The VB Err object will contain the [\\_COM MRTKTaskLib result code](#) if an error occurs.

0 if object has not expired, 1 if object has expired, or a [\\_COM MRTKTaskLib result code](#) if an error occurred

## Notes

## See also

[GetDaysRemaining](#)

See the [Object Factory tagMRTKOBJECTS Properties Summary Table](#) for a list of possible objects.

## IsUpdateAvailable

### Syntax

```
IsUpdateAvailable  
  
long IsUpdateAvailable(long *nUpdateAvail)
```

### Description

Checks if a software update is available.

### Parameters

None.

*nUpdateAvail*

Will return 1 if an update is available, 0 if it is not available.

### Return values

1 if an update is available, 0 if it is not available (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Checks the data file folder that contains the Address.cas file for an updated version of BCC Architect.

If the most current update has been previously run then this function will return 0.

The current Address.cas file location can be retrieved using the global property, mrtkDATAFILE\_LOCATION.

### See also

- [RunUpdate](#)
- [mrtkDATAFILE\\_LOCATION](#)

## RunUpdate

### Syntax

```
RunUpdate  
  
long RunUpdate()
```

## Description

Update the BCC Architect controls to the latest version.

## Parameters

None.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

## Notes

Only call RunUpdate if IsUpdateAvailable returns True.

The updater may need to reboot the computer if a file that needs to be updated is locked by the operating system. In this event, RunUpdate returns the result code E\_MRTK\_UPDATE\_RESTART\_NEEDED (0x80040417L), which indicates that a system reboot is necessary. These files will be freed after a reboot is performed.

## See also

[IsUpdateAvailable](#)

## Object Factory Properties

The MRTKObjFactory properties are defined below. After you have added a reference to the MRTKFact.dll, you can begin using these properties in your project.

### DataFilePath

#### Syntax

```
DataFilePath(nCountryID)
```

```
long DataFilePath(tagMRTKCOUNTRYIDS nCountryID, BSTR *pVal) /*  
retrieving */
```

```
long DataFilePath(tagMRTKCOUNTRYIDS nCountryID, BSTR newVal) /* set-  
ting */
```

#### Description

Sets and retrieves the location of the data files.

## Data Type

String

BSTR

## Parameters

nCountryID as tagMRTKCOUNTRYIDS – The country whose data file path you want to set or retrieve.

nCountryID – The country whose data file path you want to set or retrieve

pVal – Returns property value

newVal – Value to assign to the property

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

This property acts the same as the MRTKObjFactory.FileSearchPath property when the nCountryID is UNITED\_STATES.

This property is to be used for US, CANADA, and UK data sets.

When setting this property, you can use any of the following formats:

```
"C:\BCC Architect\Data file"
```

```
"C:\BCC Architect\Data file\"
```

```
"C:\BCC Architect\Data file\Address.cas" (United States)
```

When retrieving this property, it will always return the path to the data file folder in uppercase letters followed by a backslash, regardless of the format used to set it:

```
"C:\SATORI ARCHITECT\DATAFILE\"
```

## See also

See the [Object Factory Country IDs Summary Table](#) for a list of available countries.

## FileSearchPath

### Syntax

```
FileSearchPath  
  
long FileSearchPath(BSTR *pVal) /* retrieving */  
long FileSearchPath(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the location of the data files (United States only).

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### Notes

You can read and set this property.

Works only with BCC Architect United States. For other countries, use the DataFilePath property.

### See also

[DataFilePath](#)

## MRTKRegistrationKey

### Syntax

```
MRTKRegistrationKey
```

```
long MRTKRegistrationKey(BSTR *pVal) /*when retrieving*/
long MRTKRegistrationKey(BSTR newVal) /*when setting*/
```

#### Description

Sets and retrieves the BCC Architect registration serial number.

#### Data Type

String

BSTR

#### Parameters

#### None

pVal – Returns property value.

newVal – Value to assign to the property.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

#### Notes

You can read and set this property.

Setting this property will change the registration serial number in the registry.

## Object Factory Object Names Summary Table

Below are the string IDs of the BCC Architect objects. We recommended that you create all Task objects through the MRTKObjFactory object.

String Object Name	Description
MRTKAU.AUBatchReportTask.1	The name of the AUBatchReportTask object (Australia)
MRTKAU.AUBatchTask.1	The name of the AUBatchTask object (Australia batch address correction)

String Object Name	Description
MRTKAU.AUPostcodeTask.1	The name of the AUPostCodeTask object (Australia single address correction)
MRTKAU.AUSortTask.1	The name of the AUSortTask object (Australia)
MRTKCA.CABatchReportTask.1	The name of the CABatchReportTask object (Canada)
MRTKCA.CABatchTask.1	The name of the CABatchTask object (Canada batch address corrections)
MRTKCA.CAPostalCodeTask.1	The name of the CAPostalCodeTask object (Canada single address correction)
MRTKCA.CASortTask.1	The name of the CASortTask object (Canada)
MRTKFactory.MRTKObjFactory.1	The name of the MRTKObjFactory object
MRTKTask.CASSReportTask.1	The name of the CASSReportTask object (United States)
MRTKTask.CASSTask.1	The name of the CASSTask object (United States batch address correction)
MRTKTask.MOVETask.1	The name of the MoveTask object (United States)
MRTKTask.PresortTask.1	The name of the PresortTask object (United States)
MRTKTask.ReviewErrorsTask.1	The name of the ReviewErrorsTask object (United States)
MRTKTask.ZIPTask.1	The name of the ZIPTask object (United States single address correction)
MRTKTask.GeocodeTask.1	The name of the GeocodeTask object (United States)
MRTKUK.UKBatchReportTask.1	The name of the UKBatchReportTask object (United Kingdom)
MRTKUK.UKBatchTask.1	The name of the UKBatchTask object (United Kingdom batch address correction)
MRTKUK.UKGeocodeTask.1	The name of the UKGeocodeTask object (United Kingdom)

String Object Name	Description
MRTKUK.UKPostCodeTask.1	The name of the UKPostCodeTask object (United Kingdom single address correction)
MRTKUK.UKSortTask.1	The name of the UKSortTask object (United Kingdom)

## Object Factory tagMRTKOBJECTS Properties Summary Table

Below are the MRTKFACTORYLib enum names and values.

tagMRTKOBJECTS	Enum Value	Description
ENTERPRISE_AUBATCHREPORTTASK	28	Creates the AUBATCHREPORTTASK interface
ENTERPRISE_AUBATCHTASK	27	Creates the AUBATCHTASK interface
ENTERPRISE_AUPOSTCODETASK	25	Creates the AUPOSTCODETASK interface
ENTERPRISE_AUPRESORTTASK	26	Creates the AUPRESORTTASK interface
ENTERPRISE_CABATCHREPORTTASK	16	Creates the CABATCHREPORTTASK interface
ENTERPRISE_CABATCHTASK	15	Creates the CABATCHTASK interface
ENTERPRISE_CAPOSTCODETASK	12	Creates the CAPOSTALCODETASK interface
ENTERPRISE_CASORTTASK	14	Creates the CASORTTASK interface
ENTERPRISE_CASSREPORTTASK	4	Creates the CASSREPORTTASK interface
ENTERPRISE_CASSTASK	3	Creates the CASSTASK interface
ENTERPRISE_DATABROWSER	8	Creates the DATABROWSER interface
ENTERPRISE_GEOCODETASK	38	Creates the GEOCODETASK interface

tagMRTKOBJECTS	Enum Value	Description
ENTERPRISE_MOVETASK	24	Creates the MOVETASK interface
ENTERPRISE_PRESORTTASK	5	Creates the PRESORTTASK interface
ENTERPRISE_REVIEWERRORSTASK	9	Creates the REVIEWERRORSTASK interface
ENTERPRISE_UKBATCHREPORTTASK	17	Creates the UKBATCHREPORTTASK interface
ENTERPRISE_UKBATCHTASK	13	Creates the UKBATCHTASK interface
ENTERPRISE_UKGEOCODETASK	37	Creates the UKGEOCODETASK interface
ENTERPRISE_UKPOSTCODETASK	11	Creates the UKPOSTCODETASK interface
ENTERPRISE_UKSORTTASK	10	Creates the UKSORTTASK interface
ENTERPRISE_ZIPTASK	6	Creates the ZIPTASK interface
SMALLBUSINESS_CABATCHAGENT	23	Creates CA Batch Agent interface
SMALLBUSINESS_CAPOSTALCODEAGENT	22	Creates CA Postcode Agent interface
SMALLBUSINESS_CASORTAGENT	21	Creates CA Sort Agent interface
SMALLBUSINESS_AUBATCHAGENT	30	Creates AU Batch Agent interface
SMALLBUSINESS_AUPOSTCODEAGENT	29	Creates AU Postcode Agent interface
SMALLBUSINESS_AUPRESORTAGENT	31	Creates AU Presort Agent interface
SMALLBUSINESS_CASSAGENT	0	Creates CASS Agent interface
SMALLBUSINESS_MOVEAGENT	33	Creates MOVE Agent interface
SMALLBUSINESS_PRESORTAGENT	32	Creates Presort Agent interface

tagMRTKOBJECTS	Enum Value	Description
SMALLBUSINESS_UKBATCHAGENT	20	Creates UK Batch Agent interface
SMALLBUSINESS_UKPOSTCODEAGENT	19	Creates UK Postcode Agent interface
SMALLBUSINESS_UKSORTAGENT	18	Creates UK Mailsort Agent interface
SMALLBUSINESS_ZIPAGENT	1	Creates ZIP Agent interface
SMALLBUSINESS_ZIPBROWSER	2	Creates ZIP Browser interface

### Object Factory Country IDs Summary Table

tagMRTKCOUNTRYIDS	Enum Value	Description
AUSTRALIA	3	Australia
CANADA	2	Canada
UNITED_KINGDOM	1	United Kingdom
UNITED_STATES	0	United States of America

### Object Factory Global Properties Summary Table

Below are all the global GetProperty and SetProperty property names. The enum Property Name fields below are the values that can be passed to the GetProperty and SetProperty functions of any of the BCC Architect Tasks.

MRTK Global Prop-erties	Enum Value	Data Type	Default Value	Description
mrtkDATAFILE_LOCATION	100	String		Not currently implemented

MRTK Global Properties	Enum Value	Data Type	Default Value	Description
mrtkDATAFILE_LOCATION2	101	String		Not currently implemented
mrtkDATAFILE_LOCATION3	102	String		Not currently implemented
mrtkDELIMITER_FIELD	175	String	'\t', Chr(9) (tab)	Sets and retrieves the character that delimits a field value.
mrtkDELIMITER_RECORD	176	String	'\n', Chr(10) (line feed)	Sets and retrieves the character that delimits a record value.
mrtkFIELD_LIST_IN	200	String	EMPTY	Delimited list that specifies the input fields. You must set this field after mrtkINPUT_BLOCK_RECORD_COUNT.
mrtkFIELD_LIST_OUT	201	String	EMPTY	Delimited list that specifies the desired output fields. You must set this field after mrtkINPUT_BLOCK_RECORD_COUNT.
mrtkFIELD_LIST_OUT_ALIAS	277	String	EMPTY	No longer supported. See instead mrtkFIELD_LIST_OUT.
mrtkINPUT_BLOCK_RECORD_COUNT	202	Long	1	The number of records within block of records; you must set this field before mrtkFIELD_LIST_IN or mrtkFIELD_LIST_OUT
mrtkMAILROOM_SERVER_LIST	203	String	EMPTY	Sets and retrieves the location of the BCC Architect Server. Set this property to create a TCP/IP connection to the BCC Architect Server, which can reside on the local network or virtually anywhere. The format is: Server Name or IP Address:Port.

MRTK Global Properties	Enum Value	Data Type	Default Value	Description
mrtkPRINT_ON_SERVER	268	BOOL	FALSE	Set to TRUE to print reports on the local machine with BCC Architect Server; FALSE to print on client machine.
mrtkSECONDARY_DELIMITER_FIELD	7270	String	EMPTY	Sets and retrieves the character that delimits a field value in, for example, an address suggestion list. Useful in cases where the field that is being returned itself has delimited data. This property is currently only used for the Address Suggestion List CASSTask property. The delimiter that is set by this property cannot be the same as the delimiter set by the mrtkDELIMITER_FIELD property.
mrtkSECONDARY_DELIMITER_RECORD	7271	String	EMPTY	Sets and retrieves the character that delimits a record value in, for example, an address suggestion list. Useful in cases where the field that is being returned itself has delimited data.
mrtkSETTINGS_INI_FILE_NAME	150	String	"mrtk.ini"	The full name and path of the ini file to use.
mrtkTEMPLATE_NAME_TO_USE	151	String	EMPTY	The name of the template to use, as specified in the current ini file. Each ini file can contain more than one template. A template is a section in the ini file that stores presort settings, printer names, etc.

## The COM ZipTask Object for Correcting Single Addresses

The BCC Architect ZIPTask object validates a single address. The ZIPTask object is designed for use within address entry forms, Web pages and any other environment where single address verification is needed. After calling ZIPTask.CheckAddress(), you can retrieve the corrected address information. This includes the delivery point, postal carrier route and all the other individual elements of an address, if the address was validated. This way, you can check if an address is complete. For

example, with just a few lines of code, you can check if the user entered required unit information (e.g., suite or apartment number). If the `CheckAddress` method was unable to validate a record, you can retrieve an error code and its associated description.

Moreover, you can specify whether or not an extended error code string should be returned from `ZIPTask`. This extended error code string gives you more detail about what caused the address check to fail.

The `ZIPTask` should be created via the BCC Architect object factory (`MRTKObjFactory`).

The `ZIPTask` object can act as a client of the BCC Architect Server software. This allows the address matching process to be done on a server rather than the computer running the `ZIPTask` object. This can minimize the processing load on a workstation. Putting the `ZIPTask` in Client mode is as simple as setting the `MailRoomServer` property within the object. This property specifies where the BCC Architect Server is located and whether or not to use it. If you are using the `ZIPTask` in a Web environment, we highly recommend that you use the BCC Architect Server.

## ZIPTask Sample ASP Code

The following is a snippet of code from an ASP page that uses `ZIPTask`.

### EXAMPLE

```
Set objZIPTask = server.CreateObject("MRTKTask.ZIPTask.1")
' PrepareTask must be the first call
objZIPTask.PrepareTask
objZIPTask.AssignLOT = false
objZIPTask.AssignCounty = false

objZIPTask.ClearAddress

' Get address from calling page
objZIPTask.AddressLine1 = request.form("address")
objZIPTask.City = request.form("city")
objZIPTask.State = request.form("state")
objZIPTask.ZipCode = request.form("zip")

' Check the address based on the elements above
objZIPTask.CheckAddress

' Get the results by calling ZIPTask properties
errorcode = objZIPTask.ErrorCodes
address = objZIPTask.AddressLine1
```

```
city = objZIPTask.City
state = objZIPTask.State
zip = objZIPTask.ZipCode

deliverycheck = objZIPTask.DeliveryPointCheckDigit
carrt = objZIPTask.CarrierRoute
barcode = objZIPTask.DPBarcodeString
primary = objZIPTask.PrimaryNumber
predirectional = objZIPTask.PreDirectional
street = objZIPTask.Street
suffix = objZIPTask.Suffix
postdirectional = objZIPTask.PostDirectional
unitdesignator = objZIPTask.UnitDesignator
unitnumber = objZIPTask.UnitNumber
errorname = objZIPTask.ErrorCodeString(False)
' Must be called when you are done with the object
objZIPTask.EndTask
```

## ZIPTask Functions

The ZIPTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

### PrepareTask

#### Syntax

```
PrepareTask
long PrepareTask()
```

#### Description

Initialize and prepare the ZIPTask object before address correction.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

## Notes

PrepareTask should be called only once, when the ZIPTask object is first created.

This function must be called before any of the other functions or properties of ZIPTask. Failing to do so will cause subsequent function calls to fail.

## BrowseAddress

### Syntax

```
BrowseAddress
```

```
long BrowseAddress(long *pUpdateRequested)
```

### Description

Launches the Address Browser dialog box.

### Parameters

None.

*pUpdateRequested*

Returns 0 if the user clicks Cancel, 1 if the user clicks the Update button.

### Return values

0 if the user clicks Cancel, 1 if the user clicks the Update button (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This method will launch the Address Browser dialog box and attempt to match the address contained within the ZIPTask object. The user can then search through all the valid address ranges contained in the address files.

If you want to launch the Address Browser without an address, then call ZIPTask.ClearAddress before calling this method.

If the return value is True, then the ZIPTask object will contain the address elements for the address that the user chose to keep within the Address Browser dialog box.

By checking the ErrorCodes property after calling the CheckAddress function, you can launch the Address Browser dialog box for an address that does not validate.

See also

[CheckAddress](#)

[ClearAddress](#)

[ErrorCodes](#)

## BuildAddress

Syntax

```
BuildAddress  
long BuildAddress()
```

Description

Builds an address from its individual address elements.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

Notes

This function will build an address by combining the individual address elements, such as the primary number, street, suffix, etc., and update the AddressLine1 and AddressBlock properties.

You must set the individual element properties – listed below in the See Also section – before calling this function.

The resulting address is not matched against the USPS database nor is any formatting applied to it.

See also

[AddressBlock](#)

[AddressLine1](#)

[City](#)

[CityStateZip](#)

[PostDirectional](#)

[PreDirectional](#)

[PrimaryNumber](#)

[State](#)

[Street](#)

[Suffix](#)

[UnitDesignator](#)

[UnitNumber](#)

[ZipCode](#)

## CheckAddress

### Syntax

```
CheckAddress  
long CheckAddress()
```

### Description

Corrects, verifies and formats an address.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### Notes

This function attempts to match the address contained in the ZIPTask object against the USPS database.

After a call to `CheckAddress`, you can get the results of the process by looking at the `ErrorCode` property.

In some situations you may want to call `BrowseAddress` and launch the Address Browser if the `ErrorCode` property represents an error.

The address elements returned after calling `CheckAddress` will be formatted according to the values of the various formatting properties.

See also

[AddressBlock](#)

[AddressLine1](#)

[AddressLine2](#)

[BrowseAddress](#)

[Casing](#)

[CertifyFlag](#)

[City](#)

[CityFormat](#)

[CityStateZip](#)

[ErrorCodes](#)

[KeepAliasAddress](#)

[KeepExtraPrimaryData](#)

[PMBOutput](#)

[POBoxFormat](#)

[PostDirectional](#)

[PostDirectionalFormat](#)

[PreDirectional](#)

[PreDirectionalFormat](#)

[PrimaryAddressOutput](#)

[PrimaryNumber](#)

[RuralRouteFormat](#)

[State](#)

[Street](#)

[Suffix](#)

[SuffixFormat](#)

[UnitDesignator](#)

[UnitDesignatorFormat](#)

[UnitNumber](#)

[UnitOutput](#)

[ZipCode](#)

## ClearAddress

### Syntax

```
ClearAddress  
long ClearAddress()
```

### Description

Clears all address information contained in the ZIPTask object.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

This function can be used to bring the ZIPTask object into a known state (i.e., no address, error code, etc.) before setting the object's properties.

### See also

[CheckAddress](#)

### ParseAddress

#### Syntax

```
ParseAddress  
  
long ParseAddress()
```

#### Description

Parse an address into its individual address elements.

#### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

This function will parse the input address and return the individual address elements, such as the primary number, street, suffix, etc., formatted according to the values of the various formatting properties.

The address will not be matched against the USPS database or corrected.

See also

[AddressBlock](#)

[AddressLine1](#)

[AddressLine2](#)

[BrowseAddress](#)

[City](#)

[CityFormat](#)

[CityStateZip](#)

[POBoxFormat](#)

[PostDirectional](#)

[PostDirectionalFormat](#)

[PreDirectional](#)

[PreDirectionalFormat](#)

[PrimaryAddressOutput](#)

[PrimaryNumber](#)

[RuralRouteFormat](#)

[State](#)

[Street](#)

[Suffix](#)

[SuffixFormat](#)

[UnitDesignator](#)

[UnitDesignatorFormat](#)

[UnitNumber](#)

[ZipCode](#)

## EndTask

### Syntax

```
EndTask  
  
long EndTask()
```

### Description

Releases any system resources used by the ZIPTask processes.

### Parameters

None.

### Return values

None. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

We recommend that you call EndTask when you are done with the ZIPTask object.

You do not need to call EndTask after each call to CheckAddress.

## ZIPTask Properties

The ZIPTask properties are defined below. Once you have added a reference to the BCC Architect files, you can begin using these properties in your project.

### AddressBlock

#### Syntax

```
AddressBlock  
  
long AddressBlock(BSTR *pVal) /* retrieving */  
long AddressBlock(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the address as a complete block.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

The property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can call CheckAddress with the input address data as an address block. This process takes two or more lines of text, identifies which lines contain address data, checks the address, and then reconstructs the address block using the formatted address lines, business name and last line.

The address block must be a stream of delimited text.

The end of each line in the address block should be delimited with a carriage return/line feed, line feed, carriage return or tab.

The delimiter used must be consistent throughout the address block.

The following definitions are used to describe the lines in the block of text:

- Last Line – Contains the city, state and ZIP Code information.
- Address Line – This line is always required and specifies the actual delivery address. By USPS standards, this line should be directly above the Last Line of the address.
- Unit Designator and Number – This line only exists in the input address block and will only

be recognized if the address line does not contain a unit designator and number.

- PMB – Information for a personal mailbox. This is ignored during the correction process.

The matching process starts at the bottom-most line of the address block and then moves up until it finds a recognizable ZIP Code or city/state combination. This line will be designated as the Last Line of the address.

Once the Last Line has been located, the following schemes will be used to attempt to validate the address (listed in order of precedence). Items in brackets are optional. For Puerto Rico addresses, the business name will be treated as the urbanization if it is recognizable as such.

<Business Name>

<PMB>

<Unit Designator and Number>

Address Line

Last Line

<Business Name>

<PMB>

Address Line

<Unit Designator and Number>

Last Line

<Unit Designator and Number>

Address Line

Business Name or PMB

Last Line

Address Line

<Unit Designator and Number>

Business Name or PMB

Last Line

On output, the address block will be constructed as follows.

Unused Line 1

...

Unused Line N

<Business Name>

<Urbanization>

<PMB>

Address Line <Unit Designator and Number>

Last Line

Unused Line N+1

...

Unused Line M

All spaces at the beginning and end of each line and extra spaces between words will be removed. The output address block will be created with a carriage return/line feed as the delimiter.

Any lines of data that are not part of the corrected address will be returned in the address block. They will be placed in the reconstructed address block in the order that they were input and relative to (above/below) the address lines as they were input.

See also

[CheckAddress](#)

[Casing](#)

[KeepAliasAddress](#)

[KeepExtraPrimaryData](#)

[PMBOutput](#)

[POBoxFormat](#)

[PostDirectionalFormat](#)

[PreDirectionalFormat](#)

[PrimaryAddressOutput](#)

[RuralRouteFormat](#)

[SuffixFormat](#)

[UnitDesignatorFormat](#)

[UnitOutput](#)

## AddressInputPreference

### Syntax

```
AddressInputPreference
```

```
long AddressInputPreference(long *pVal) /* retrieving */
```

```
long AddressInputPreference(long newVal) /* setting */
```

### Description

Determines which address to give preference to if both a street address and a PO Box address are found and matched.

### Data Type

Long

long

### Parameters

#### None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### Notes

You can read and set this property.

This property only comes into play if the two addresses are input as separate address lines. If they are input on the same address line, then USPS rules dictate that the PO Box address has priority, regardless of position.

Possible values are:\

- 0 – By position, giving preference to the bottom address when there are two address lines; this is the default value
- 1 – Prefer PO Box address
- 2 – Prefer street address

See also

[CheckAddress](#)

## AddressLine1

Syntax

```
AddressLine1
```

```
long AddressLine1(BSTR *pVal) /* retrieving */
```

```
long AddressLine1(BSTR newVal) /* setting */
```

Description

Sets and retrieves top address line of an address.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

AddressLine1 is the top address line, above AddressLine2.

After calling CheckAddress, the value of the PrimaryAddressOutput property determines the location of the primary address line, if there is a single address line.

If there are two address lines, AddressLine2 will contain the primary address line and AddressLine1 will contain the secondary address information.

Both AddressLine1 and AddressLine2 are formatted according to the values of the various formatting properties.

## See also

[CheckAddress](#)

[AddressLine2](#)

[Casing](#)

[KeepAliasAddress](#)

[KeepExtraPrimaryData](#)

[PMBOutput](#)

[POBoxFormat](#)

[PostDirectionalFormat](#)

[PreDirectionalFormat](#)

[PrimaryAddressOutput](#)

[RuralRouteFormat](#)

[SuffixFormat](#)

[UnitDesignatorFormat](#)

[UnitOutput](#)

## AddressLine2

### Syntax

```
AddressLine2  
  
long AddressLine2(BSTR *pVal) /* retrieving */  
long AddressLine2(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the second address line of an address.

### Data Type

String

BSTR

### Parameters

#### None

pVal – Returns property value.

newVal - Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### Notes

You can read and set this property.

AddressLine2 is the bottom address line, below AddressLine1 and above the city, state and ZIP Code.

After calling `CheckAddress`, the value of the `PrimaryAddressOutput` property determines the location of the primary address line, if there is a single address line.

If there are two address lines, `AddressLine2` will contain the primary address line and `AddressLine1` will contain the secondary address information.

Both `AddressLine1` and `AddressLine2` are formatted according to the values of the various formatting properties.

See also

[CheckAddress](#)

[AddressLine1](#)

[Casing](#)

[KeepAliasAddress](#)

[KeepExtraPrimaryData](#)

[PMBOutput](#)

[POBoxFormat](#)

[PostDirectionalFormat](#)

[PreDirectionalFormat](#)

[PrimaryAddressOutput](#)

[RuralRouteFormat](#)

[SuffixFormat](#)

[UnitDesignatorFormat](#)

[UnitOutput](#)

`AddressLineAbbreviated`

Syntax

```
AddressLineAbbreviated
```

```
long AddressLineAbbreviated(long *pVal) /* retrieving */  
long AddressLineAbbreviated(long newVal) /* setting */
```

#### Description

Determines if the address lines are returned abbreviated.

#### Data Type

Long

long

#### Parameters

#### None

pVal - Returns 0 if False, 1 if True.

newVal - Set to 0 for False, 1 for True.

#### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

If you set this to 1, the primary address line will be abbreviated to no more than 30 characters.

You must call CheckAddress before the selected abbreviation option is applied to the address elements in the ZIPTask object.

The default value for this property is 0.

#### See also

[CheckAddress](#)

[AddressLine1](#)

[AddressLine2](#)

## AddressSuggestionList

### Syntax

```
AddressSuggestionList
long AddressSuggestionList (BSTR pVal) /* retrieving */
```

### Description

A delimited list of address suggestions that is returned when an input address is invalid and not able to be corrected.

### Data Type

String

BSTR

### Parameters

None

*pVal*

Returns property value.

### Return values

Property value. The VB Err object will contain the BCC Architect Architect result code if an error occurs.

0 if successful, otherwise a COM MRTKTaskLib result code

### Notes

Call CheckAddress before attempting to retrieve the value of this property.

The field will display a maximum of 10 entries, all of which have had their deliverability confirmed (DPV).

The following table describes the delimiters that are used by default in the output text.

Delimiter Type	ASCII	Hex
Field	25	19
Record	26	1A

The type of input addresses that will generally return suggestions are those that have certain address elements that are missing or invalid, such as street suffixes and street directionals. Suggestions return the following fields:

- AddressLine 1
- City
- State
- ZIP Code

See also

- CheckAddress
- ReturnAddressSuggestionList

## ApplyCasingBusiness

Syntax

```
ApplyCasingBusiness
```

```
long ApplyCasingBusiness(long *pVal) /* retrieving */
```

```
long ApplyCasingBusiness(long newVal) /* setting */
```

Description

Determines if the case formatting specified in the Casing property is applied to the business name.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

You must call CheckAddress before the selected casing option is applied to the business name in the ZIPTask object.

The default value for this property is 1.

### See also

[CheckAddress](#)

[Casing](#)

## AssignCounty

### Syntax

```
AssignCounty  
  
long AssignCounty(long *pVal) /* retrieving */  
long AssignCounty(long newVal) /* setting */
```

### Description

Determines if county information should be returned.

### Data Type

Long

long

### Parameters

#### None

pVal - Returns 0 if False, -1 if True.

newVal - Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

You should only set this property if you do not want county information returned for a given address.

The default value for this property is True.

### See also

[CheckAddress](#)

## AssignLOT

### Syntax

```
AssignLOT  
  
long AssignLOT(long *pVal) /* retrieving */  
long AssignLOT(long newVal) /* setting */
```

### Description

Determines whether the LOT number should be returned.

### Data Type

Long

long

### Parameters

#### None

pVal - Returns 0 if False, -1 if True.

newVal - Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### Notes

You can read and set this property.

This property should be set if you want Line-of-Travel information for an address.

The default value for this property is False.

### See also

[CheckAddress](#)

## AssignRDI

### Syntax

```
AssignRDI  
  
long AssignRDI(long *pVal) /* retrieving */  
long AssignRDI(long newVal) /* setting */
```

### Description

Determines whether residential delivery indicator data should be returned.

### Data Type

Long

long

### Parameters

None

pVal - Returns 0 if False, -1 if True.

newVal - Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

RDI is available through an add-on for BCC Architect.

To use this property, you must subscribe to the RDI service through the USPS.

### See also

[CheckAddress](#)

## BusinessName

### Syntax

```
BusinessName
```

```
long BusinessName(BSTR *pVal) /* retrieving */
```

```
long BusinessName(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the business name of an address.

### Data Type

String

BSTR

### Parameters

None

pVal - Returns property value.

newVal - Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

## See also

[CheckAddress](#)

[ApplyCasingBusiness](#)

[Casing](#)

## Capitalize

### Syntax

```
Capitalize  
long Capitalize(long *pVal) /* retrieving */  
long Capitalize(long newVal) /* setting */
```

### Description

Determines if the returned address elements will be capitalized.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal - Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

You must call `CheckAddress` before the address elements in the `ZIPTask` object will be capitalized.

This property has been superseded by the more flexible `Casing` property, but has been retained for backward compatibility.

The default value for this property is `False`.

#### See also

[CheckAddress](#)

[Casing](#)

## CarrierRoute

#### Syntax

```
CarrierRoute  
  
long CarrierRoute(BSTR *pVal) /* retrieving */  
long CarrierRoute(BSTR newVal) /* setting */
```

#### Description

Sets and retrieves the carrier route data for an address.

#### Data Type

String

BSTR

#### Parameters

None

`pVal` – Returns property value.

`newVal` – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

### See also

[CheckAddress](#)

## CarrtCoded

### Syntax

```
CarrtCoded  
  
long CarrtCoded(long *pVal) /* retrieving */
```

### Description

Indicates if carrier route information was successfully assigned to this address.

### Data Type

Long

long

### Parameters

None

pVal - Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

## See also

[CheckAddress](#)

## Casing

### Syntax

```
Casing  
  
long Casing(long *pVal) /* retrieving */  
long Casing(long newVal) /* setting */
```

### Description

Determines the casing format applied to returned address elements.

### Data Type

Long

long

### Parameters

None

`pVal` – Returns property value.

`newVal` – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call `CheckAddress` before the selected casing option is applied to the address elements in the `ZIPTask` object.

Possible values are:

- 0 – Upper case
- 1 – Lower case
- 2 – Mixed case (default value)

See also

[CheckAddress](#)

[ApplyCasingBusiness](#)

[Capitalize](#)

## CASSDate

Syntax

VB:

`CASSDate`

```
long CASSDate(long *pVal) /* retrieving */
```

```
long CASSDate(long newVal) /* setting */
```

Description

Sets and retrieves the date that this address was last corrected.

Data Type

Long

long

Parameters

None

`pVal` - Returns property value.

`newVal` - Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

CASSdate is a binary field that contains information about the last results from the address matching engine.

One of the fields within the CASSDate is the issue number that was used to validate the address.

If you are going to batch process, we strongly recommend that you save this field because it will allow a batch process (via CASSTask, for instance) to skip this record if it was previously corrected, based on the CertifyFlag property.

### See also

[CheckAddress](#)

[CertifyFlag](#)

## CensusBlock

### Syntax

```
CensusBlock  
long CensusBlock(BSTR *pVal)
```

### Description

Retrieves the census block data for an address.

### Data Type

String

BSTR

### Parameters

None

pVal - Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property requires the Geocode add-on.

### See also

[CheckAddress](#)

[CensusTract](#)

[GeocodeFootnote](#)

[Latitude](#)

[Longitude](#)

[MSACode](#)

[UseGeocode](#)

## CensusTract

### Syntax

```
CensusTract
```

```
long CensusTract (BSTR *pVal)
```

### Description

Retrieves the census tract data for an address.

### Data Type

String

BSTR

## Parameters

None

pVal – Returns property value

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property requires the Geocode add-on.

## See also

[CheckAddress](#)

[CensusBlock](#)

[GeocodeFootnote](#)

[Latitude](#)

[Longitude](#)

[MSACode](#)

[UseGeocode](#)

## CertifyFlag

### Syntax

```
CertifyFlag
```

```
long CertifyFlag(long *pVal) /* retrieving */
```

```
long CertifyFlag(long newVal) /* setting */
```

## Description

Specifies whether a previously corrected address should be rechecked.

## Data Type

Long

long

## Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read or set this property.

To speed up processing of a batch of addresses, set this property to 1 and `ZIPTask.CheckAddress()` will skip any addresses that have been previously corrected with the current issue. The `CASSDate` property must be set in order for an address to be skipped.

Possible values are:

- 0 – Skip address if previously corrected with the current issue.
- 1 – Always check address.

## See also

[CheckAddress](#)

[CASSDate](#)

## City

### Syntax

```
City  
  
long City(BSTR *pVal) /* retrieving */  
long City(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the city of an address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

### See also

[CheckAddress](#)

[Casing](#)

[CityFormat](#)

## CityFormat

### Syntax

```
CityFormat  
  
long CityFormat(long *pVal) /* retrieving */  
long CityFormat(long newVal) /* setting */
```

### Description

Determines how the returned city value will be abbreviated, if at all.

### Data Type

Long

long

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return full city name (default value)
- 1 – Return abbreviated city name (if one exists) of no more than 13 characters
- 2 – Return as input. THIS VALUE IS NO LONGER SUPPORTED.

See also

[CheckAddress](#)

[AddressBlock](#)

[City](#)

[CityStateZip](#)

## CityStateZip

Syntax

```
CityStateZip  
long CityStateZip(BSTR *pVal) /* retrieving */  
long CityStateZip(BSTR newVal) /* setting */
```

Description

Sets and retrieves the city, state and ZIP Code of an address.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can read and set this property.

See also

[CheckAddress](#)

[CityFormat](#)

## CongressionalDistrict

Syntax

```
CongressionalDistrict  
long CongressionalDistrict(BSTR *pVal)
```

Description

Retrieves the congressional district data for an address.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call [CheckAddress](#) before attempting to retrieve the value of this property.

You must have the Geocode add-on to generate congressional district data.

See also

[CheckAddress](#)

## CountyCode

### Syntax

```
CountyCode  
long CountyCode(long *pVal)
```

### Description

Retrieves the county FIPS code associated with a ZIP Code as an integer.

### Data Type

Long

long

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Read-only property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property contains the full 5-digit County FIPS code, where the first two digits represent the state and the last three digits represent the county.

### See also

[CheckAddress](#)

[CountyFIPSCode](#)

## CountyFIPSCode

### Syntax

```
CountyFIPSCode  
  
long CountyFIPSCode (BSTR *pVal)
```

### Description

Retrieves the county FIPS code associated with ZIP Code as a string.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Read-only property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property contains the full 5-digit County FIPS code, where the first two digits represent the state and the last three digits represent the county.

### See also

[CheckAddress](#)

[CountyCode](#)

## CountyName

### Syntax

```
CountyName  
long CountyName(BSTR *pVal)
```

### Description

Retrieves the county name associated with this address' ZIP Code.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

### See also

[CheckAddress](#)

## DatafileLocation

### Syntax

```
DatafileLocation  
long DatafileLocation(BSTR *pVal) /* retrieving */  
long DatafileLocation(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the location of the address correction data files.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

This property stores the full path name of the folder containing the Address.cas file.

## DeliveryPointCheckDigit

### Syntax

```
DeliveryPointCheckDigit  
long DeliveryPointCheckDigit(BSTR *pVal)
```

### Description

Retrieves the delivery point/check digit of a corrected address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

#### See also

[CheckAddress](#)

## DPBarcodeString

#### Syntax

```
DPBarcodeString  
long DPBarcodeString(BSTR *pVal)
```

#### Description

Retrieves the delivery point barcode for a corrected address.

#### Data Type

String

BSTR

#### Parameters

None

pVal – Returns property value.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

You must have the `SatBar.ttf` font installed before you can properly print or display this barcode.

## See also

[CheckAddress](#)

## DPVCoded

### Syntax

`DPVCoded`

```
long DPVCoded(long *pVal)
```

### Description

Indicates if the address was matched to a known delivery point.

### Data Type

Long

long

### Parameters

None

`pVal` – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

See also

[CheckAddress](#)

[UseDPV](#)

## DPVFailureAsError

Syntax

```
DPVFailureAsError  
  
long DPVFailureAsError(long *pVal) /* retrieving */  
long DPVFailureAsError(long newVal) /* setting */
```

Description

Determines how to handle addresses that do not pass secondary DPV processing.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can read or set this property.

Set to 1 to treat addresses that have a valid primary but fail DPV because of missing or invalid unit information as uncorrected addresses. The ZIP+4 Code will not be returned for these records.

The default value for this property is 0.

See also

[CheckAddress](#)

## DPVFootnotes

Syntax

```
DPVFootnotes
long DPVFootnotes (BSTR *pVal)
```

Description

Returns additional information from a DPV lookup.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call [CheckAddress](#) before attempting to retrieve the value of this property.

This property returns a code indicating specific information about the DPV lookup, as described in the following table.

Value	Description
LK	Processing locked out due to a seed record being processed.

Value	Description
AA	Matched to the ZIP+4 file.
A1	No match against the ZIP+4 file.
BB	Matched to DPV file (all components confirmed).
CC	Matched only after removing secondary information, they were presented but invalid.
N1	Input Primary matched, but high-rise missing secondary number.
M1	Primary number missing.
M3	Primary number invalid.
P1	Input missing PO, RR, HC box number.
P3	Failed DPV because of invalid PO, RR, or HC box number
RR	Matched CMRA (found in CMRA file).
R1	Matched CMRA, but secondary number (i.e., PMB) missing.
U1	Matched unique zip code.
G1	Matched General delivery.
F1	Matched military address.

See also

[CheckAddress](#)

[UseDPV](#)

## DPVIndicator

Syntax

```
DPVIndicator
```

```
long DPVIndicator(BSTR *pVal)
```

### Description

Indicates the results of DPV processing.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property returns a single character indicating the results of DPV processing, as described in the following table.

Value	Description
Y	Both the primary and secondary (if present) address information validated against the DPV database.
S	The primary address is valid according to DPV, but the secondary is invalid.
D	The primary address is valid according to DPV, but the address is missing secondary information.
N	The primary address is not valid according to DPV.

Value	Description
""	The address was not matched to the DPV table because it lacks components needed for the lookup. This usually means the record is not ZIP+4 Coded.
X	The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.
E	The DPV data file is more than 105 days old. By USPS restrictions, no more addresses can be presented to the DPV table.

See also

[CheckAddress](#)

[UseDPV](#)

## DPVIsCMRA

Syntax

```
DPVIsCMRA
long DPVIsCMRA(long *pVal)
```

Description

Indicates if the address is a commercial mail-receiving agent.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

The returned property value will always be 0 if the address is not DPV coded.

### See also

[CheckAddress](#)

[UseDPV](#)

## DPVIsDoorNotAccessible

### Syntax

```
DPVIsDoorNotAccessible  
  
long DPVIsDoorNotAccessible(long *pVal)
```

### Description

Indicates if address is one where carriers cannot knock on the door to deliver mail that will not fit into a mailbox or where carriers cannot physically access a residence/building.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

C++:

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Examples of addresses that fall into this category include those within a gated community.

This is a read-only property.

Call `CheckAddress` before attempting to retrieve the value of this property.

The returned property value will always be 0 if the address is not DPV coded.

### See also

[CheckAddress](#)

[UseDPV](#)

## DPVIsNoSecureLocation

### Syntax

```
DPVIsNoSecureLocation  
  
long DPVIsNoSecureLocation(long *pVal)
```

### Description

Indicates if the location of the address is unsecure. The USPS can access the door, but cannot leave a package due to security concerns.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

#### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

C++:

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This is a read-only property.

Call CheckAddress before attempting to retrieve the value of this property.

The returned property value will always be 0 if the address is not DPV coded.

#### See also

[CheckAddress](#)

[UseDPV](#)

## DPVIsNoStat

#### Syntax

```
DPVIsNoStat  
  
long DPVIsNoStat(long *pVal)
```

#### Description

Indicates if the address is not receiving delivery and is not counted as a possible delivery. The address is not receiving delivery because: 1) delivery has not been established, 2) The customer receives mail as part of a drop, 3) the carrier destroys or returns all of the mail.

#### Data Type

Long

long

#### Parameters

None

pVal – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

C++:

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

### See also

[CheckAddress](#)

[UseDPV](#)

## DPVIsThrowback

### Syntax

```
DPVIsThrowback  
long DPVIsThrowback(long *pVal)
```

### Description

Indicates if the address is a street address; however, the deliver is made to the customer's P.O. Box address.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

C++:

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This is a read-only property.

Call `CheckAddress` before attempting to retrieve the value of this property.

The returned property value will always be 0 if the address is not DPV coded.

#### See also

[CheckAddress](#)

[UseDPV](#)

## DPVIsVacant

#### Syntax

```
DPVIsVacant  
long DPVIsVacant(long *pVal)
```

#### Description

Indicates if the address is unoccupied.

#### Data Type

Long

long

#### Parameters

None

pVal – Returns 0 if False, -1 if True.

#### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

The returned property value will always be 0 if the address is not DPV coded.

See also

[CheckAddress](#)

[UseDPV](#)

## DPVLocation

### Syntax

```
DPVLocation  
  
long DPVLocation(BSTR *pVal) /* retrieving */  
long DPVLocation(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the location of the DPV data file.

### Data Type

String

BSTR

### Parameters

None

`pVal` – Returns property value.

`newVal` – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

The format of the returned property value is the full path name of the folder that contains the DPV.cas file, ending with a backslash.

## DPVResolveMultipleResponse

### Syntax

```
DPVResolveMultipleResponse  
  
long DPVResolveMultipleResponse(long *pVal) /* retrieving */  
long DPVResolveMultipleResponse(long newVal)/* setting */
```

### Description

Determines if DPV should be used to resolve multiple address matches.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Set to True to use DPV to help resolve multiple responses when address matching.

The default value for this property is 0.

### See also

[CheckAddress](#)

## ErrorCodes

### Syntax

```
ErrorCodes  
  
long ErrorCodes(long *pVal)
```

### Description

Retrieves the error code from an address-matching operation.

### Data Type

Long

long

### Parameters

None

pVal - Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

See the MRTKTaskLib Error Codes Table for a list of error code values and their descriptions.

### See also

[CheckAddress](#)

[ErrorCodeString](#)

## ErrorCodeString

### Syntax

```
ErrorCodeString(nExtendedError)
```

```
long ErrorCodeString(long nExtendedError, BSTR *pVal)
```

#### Description

Retrieves a description of the results of the address correction process.

#### Data Type

String

BSTR

#### Parameters

nExtendedError – Set this to 0 to return the standard description, 1 to return an extended description (as Long).

nExtendedError – Set this to 0 to return the standard description, 1 to return an extended description.

pVal – Returns property value.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

#### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

#### See also

[CheckAddress](#)

[ErrorCodes](#)

#### ExtraInfo

#### Syntax

```
ExtraInfo
```

```
long ExtraInfo(BSTR *pVal)
```

## Description

Retrieves extra information in an address that was unable to be parsed as another address field.

## Data Type

String

BSTR

## Parameters

None

pVal – Returns property value

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

## See also

[CheckAddress](#)

## FirmOutput

### Syntax

```
FirmOutput
```

```
long FirmOutput(long *pVal) /* retrieving */
```

```
long FirmOutput(long newVal) /* setting */
```

### Description

Determines where ZIPTask places the business name in a corrected address if it is not input as BusinessName.

## Data Type

Long

long

## Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call CheckAddress before the selected option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return as input (default value).
- 1 – Return as BusinessName (only if input BusinessName was blank).
- 2 – Always return as BusinessName (will overwrite input BusinessName).

## See also

[CheckAddress](#)

[BusinessName](#)

## GeocodeFootnote

### Syntax

```
GeocodeFootnote
```

```
long GeocodeFootnote (BSTR *pVal)
```

#### Description

Indicates how specific the geocode information is for this address.

#### Data Type

String

BSTR

#### Parameters

None

pVal – Returns property value.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can only read this property.

This property requires the Geocode add-on.

Call CheckAddress before attempting to retrieve the value of this property.

This property returns a code that indicates the granularity of ZIP Code match for a geocode lookup, as described in the following table.

Value	Description
00	Unable to perform a Geocode lookup.
03	Geocode data based on a 3-digit ZIP Code.
05	Geocode data based on a 5-digit ZIP Code.
07	Geocode data based on a 7-digit ZIP Code.
09	Geocode data based on a 9-digit ZIP Code.

See also

[CheckAddress](#)

[UseGeocode](#)

## HighwayContractFormat

Syntax

```
HighwayContractFormat  
  
long HighwayContractFormat(long *pVal) /* retrieving */  
long HighwayContractFormat(long newVal) /* setting */
```

Description

Determines the format of corrected highway contract address.

Data Type

Long

long

Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can read and set this property.

You must call CheckAddress before the selected option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is the default value)
- 1 – Return standardized USPS abbreviation with punctuation
- 2 – Return full word(s)

See also

[CheckAddress](#)

## IsResidence

Syntax

```
IsResidence  
long IsResidence(long *pVal)
```

Description

Indicates if the address is residential.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property requires the RDI add-on.

To use this property, you must subscribe to the Residential Delivery Indicator service through the USPS.

See also

[CheckAddress](#)

[AssignRDI](#)

## KeepAliasAddress

Syntax

```
KeepAliasAddress
```

```
long KeepAliasAddress(long *pVal) /* retrieving */
```

```
long KeepAliasAddress(long newVal) /* setting */
```

Description

Determines whether to return the street name alias as specified in the input or the official street name.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can read and set this property.

If property value is True, the input street name alias is returned.

Otherwise, the official street name from the USPS database is returned.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPTask` object.

The default value for this property is True.

See also

### [CheckAddress](#)

## KeepExtraPrimaryData

### Syntax

```
KeepExtraPrimaryData  
  
long KeepExtraPrimaryData(long *pVal) /* retrieving */  
long KeepExtraPrimaryData(long newVal) /* setting */
```

### Description

Determines if `ZIPTask` preserves extra information found in the primary address line.

### Data Type

Long

long

### Parameters

None

`pVal` – Returns 0 if False, -1 if True.

`newVal` – Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call `CheckAddress` before this option is applied to the address elements in the `ZIPTask` object.

The default value for this property is `False`.

See also

### [CheckAddress](#)

## KeepNonMailingCity

### Syntax

```
KeepNonMailingCity  
  
long KeepNonMailingCity(long *pVal) /* retrieving */  
long KeepNonMailingCity(long newVal) /* setting */
```

### Description

Determines whether to return the non-mailing city as specified in the input or as the preferred city name.

### Data Type

Long

long

### Parameters

None

`pVal` – Returns 0 if `False`, -1 if `True`.

`newVal` – Set to 0 for `False`, non-zero value for `True`.

### Return values

0 if `False`, -1 if `True`. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

If property value is True, a non-mailing city name is returned input. Otherwise, the USPS preferred city name is returned.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPTask` object.

## See also

[CheckAddress](#)

[City](#)

[CityStateZip](#)

## LACS

### Syntax

LACS

```
long LACS(long *pVal)
```

### Description

Indicates if address was identified and changed by the LACS<sup>Link</sup> process.

### Data Type

Long

long

### Parameters

None

`pVal` – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

## See also

[CheckAddress](#)

## LACSFootnote

### Syntax

```
LACSFootnote  
  
long LACSFootnote(BSTR *pVal)
```

### Description

Returns additional information about a LACS<sup>Link</sup> lookup.

### Data Type

String

BSTR

### Parameters

None

`pVal` – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property returns a code that indicates the results about a LACS<sup>Link</sup> lookup, as described in the following table:

Value	Description
""	Not processed / Seed record.
00	No match.
09	Matched to default high-rise address; address not updated.
14	Match failure to build new address.
92	Match secondary dropped from input.
A	Match success.

See also

[CheckAddress](#)

[UseLACS](#)

[LACSIndicator](#)

## LACSIndicator

Syntax

```
LACSIndicator
```

```
long LACSIndicator(BSTR *pVal)
```

Description

Returns the result of a LACS<sup>Link</sup> lookup.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property returns a single character that indicates the result of a LACS<sup>Link</sup> lookup, as described in the following table.

Value	Description
""	Not processed.
N	No match / Matched, but there was a failure to build new address.
Y	Match success.
S	Match with secondary dropped from input.
F	Seed record.

## See also

[CheckAddress](#)

[UseLACS](#)

[LACSIndicator](#)

## Latitude

## Syntax

Latitude

```
long Latitude(BSTR *pVal)
```

## Description

Retrieves the latitude coordinate for an address.

## Data Type

String

BSTR

## Parameters

None

pVal – Returns property value.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property requires the Geocode add-on.

## See also

[CheckAddress](#)

[CensusBlock](#)

[CensusTract](#)

[GeocodeFootnote](#)

[Longitude](#)

[MSACode](#)

[UseGeocode](#)

## Longitude

### Syntax

```
Longitude
```

```
long Longitude (BSTR *pVal)
```

#### Description

Retrieves the longitude coordinate for an address.

#### Data Type

String

BSTR

#### Parameters

None

pVal – Returns property value.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property requires the Geocode add-on.

#### See also

[CheckAddress](#)

[CensusBlock](#)

[CensusTract](#)

[GeocodeFootnote](#)

[Latitude](#)

[MSACode](#)

[UseGeocode](#)

## LOTNumber

### Syntax

```
LotNumber  
long LotNumber (BSTR *pVal)
```

### Description

Retrieves the line-of-travel number for an address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

### See also

[CheckAddress](#)

## MSACode

### Syntax

```
MSACode  
  
long MSACode (BSTR *pVal)
```

### Description

Retrieves the metropolitan statistical area code for an address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property requires the Geocode add-on.

### See also

[CheckAddress](#)

[CensusBlock](#)

[CensusTract](#)

[GeocodeFootnote](#)

[Latitude](#)

[Longitude](#)

[UseGeocode](#)

## MailRoomServer

### Syntax

```
MailRoomServer  
  
long MailRoomServer(BSTR *pVal) /* retrieving */  
long MailRoomServer(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the BCC Architect Server IP address and port number.

### Data Type

String

C++:

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Use this property to set and retrieve the location of the BCC Architect Server.

Setting this property creates a TCP/IP connection to the BCC Architect Server, which can reside on the local network or virtually anywhere.

We recommend that the BCC Architect Server be used when validating addresses from a Web site.

The format is: [Server Name or IP Address]:[Port].

Currently, going outside of the proxy server might not be supported.

See also

["BCC Architect Server" on page 3](#)

## MatchedToDefault

### Syntax

```
MatchedToDefault
```

```
long MatchedToDefault(long *pVal)
```

### Description

Indicates if an address was matched to default ZIP Code.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property will be True if multiple responses exist and the address is matched to the default entry because of missing secondary information.

See also

[CheckAddress](#)

## Plus4Coded

Syntax

```
Plus4Coded  
long Plus4Coded(long *pVal)
```

Description

Indicates if an address was corrected and assigned a ZIP+4 Code.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

See also

[CheckAddress](#)

## PMB

### Syntax

PMB

```
long PMB(BSTR *pVal)
```

### Description

Retrieves the private mailbox number of an address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

### See also

[CheckAddress](#)

[PMBOutput](#)

## PMBOutput

### Syntax

PMBOutput

```
long PMBOutput(long *pVal) /* retrieving */
```

```
long PMBOutput(long newVal) /* setting */
```

#### Description

Determines the output location of a private mailbox number.

#### Data Type

Long

long

#### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPTask` object.

Possible values are:

- 0 – Return on same line as unit information (default value)
- 1 – Return as input

#### See also

[CheckAddress](#)

[PMB](#)

## POBoxDeliveryOnlyZip

### Syntax

```
POBoxDeliveryOnlyZip  
Int POBoxDeliveryOnlyZip
```

### Description

Indicates whether delivery is to an address in a PO Box only zone.

### Data Type

Integer

### Parameters

None

### Return Values

Returns 0 if False, -1 if True.

## POBoxFormat

### Syntax

```
POBoxFormat  
long POBoxFormat(long *pVal) /* retrieving */  
long POBoxFormat(long newVal) /* setting */
```

### Description

Determines the format of corrected PO Box addresses.

### Data Type

Long

long

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPTask` object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is also the default value).
- 1 – Return standardized USPS abbreviation with punctuation.
- 2 – Return full words.

## See also

[CheckAddress](#)

## PostDirectional

### Syntax

```
PostDirectional  
long PostDirectional(BSTR *pVal)
```

### Description

Retrieves the post-directional for a corrected address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can only read this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

A post-directional is a street direction indicator that follows the street name, such as the "E" in "Alaska Ave E."

#### See also

[CheckAddress](#)

## PostDirectionalFormat

#### Syntax

```
PostDirectionalFormat  
  
long PostDirectionalFormat(long *pVal) /* retrieving */  
long PostDirectionalFormat(long newVal) /* setting */
```

#### Description

Determines the format of the post-directional for a corrected address.

#### Data Type

Long

long

#### Parameters

None

- pVal – Returns property value.
- newVal – Value to assign to the property.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
- 1 – Return standardized USPS abbreviation with punctuation
- 2 – Return full word(s)

#### See also

[CheckAddress](#)

[PostDirectional](#)

#### PreDirectional

##### Syntax

```
PreDirectional  
long PreDirectional(BSTR *pVal)
```

##### Description

Retrieves the pre-directional of a corrected address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

A pre-directional is a street direction indicator that precedes the street name, such as the "W" in "W 6th Street."

### See also

[CheckAddress](#)

## PreDirectionalFormat

### Syntax

```
PreDirectionalFormat  
long PreDirectionalFormat(long *pVal) /* retrieving */  
long PreDirectionalFormat(long newVal) /* setting */
```

### Description

Determines the format of the pre-directional of a corrected address.

### Data Type

Long

long

#### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is also the default value).
- 1 – Return standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

#### See also

[CheckAddress](#)

[PreDirectional](#)

#### PrimaryAddressOutput

##### Syntax

```
PrimaryAddressOutput
```

```
long PrimaryAddressOutput(long *pVal) /* retrieving */
```

```
long PrimaryAddressOutput(long newVal) /* setting */
```

## Description

Determines on which line the corrected address will be returned.

## Data Type

Long

long

## Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPTask` object.

Possible values are:

- 0 – Return as `AddressLine1` (if `AddressLine2` is blank; otherwise, the primary address line will be returned as `AddressLine2` with the secondary address data in `AddressLine1`); this is the default value.
- 1 – Return as `AddressLine2`.

## See also

[CheckAddress](#)

[AddressLine1](#)

[AddressLine2](#)

## PrimaryNumber

### Syntax

```
PrimaryNumber  
  
long PrimaryNumber(long *pVal) /* retrieving */  
long PrimaryNumber(long newVal) /* setting */
```

### Description

Sets and retrieves the primary street number of an address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

A primary number is the house number that precedes the street name, such as the "2815" in "2815 2nd Ave."

### See also

[CheckAddress](#)

## RecordType

### Syntax

```
RecordType  
  
long RecordType(BSTR *pVal) /* retrieving */
```

### Description

Retrieves the type for an address record.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property returns a single letter that represents the type of address that is contained in the ZIPTask object.

For example, a PO BOX address will return a value of "P."

The following record types are currently valid:

- S – Street record
- P – Post Office box
- R – Rural Route or Highway Contract

- H – High-rise, Building, or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

See also

[CheckAddress](#)

## ReturnAddressSuggestionList

Syntax

Description

Indicates whether to return a list of suggested addresses for an input address that is invalid and can't be corrected.

Data Type

Boolean

BOOL

Parameters

None

*pVal*

Returns property value.

*newVal*

Value

Return values

The default value is False.

Notes

You can read and set this property.

This property should be set after calling `PrepareTask`, but before `ValidateProperties`.

This property should be set to True if you want to turn on the AddressSuggestionList property, which provides a list of suggested addresses returned for an input address that is invalid.

See also

- CheckAddress
- AddressSuggestionList

## ReturnInputAddressOnUnconfirmedDPV

Syntax

```
ReturnInputAddressOnUnconfirmedDPV  
  
long ReturnInputAddressOnUnconfirmedDPV(long *pVal) /* retrieving */  
long ReturnInputAddressOnUnconfirmedDPV(long newVal) /* setting */
```

Description

Determines whether to roll back corrected addresses that are not DPV confirmable.

Data Type

Long  
long

Parameters

*pVal*

Returns 0 if False, -1 if True.

Return Values

0 if False, -1 if True. The VB Err object will contain the COM MRTKTaskLib result code if an error occurs.

0 if successful, otherwise a COM MRTKTaskLib result code.

Notes

The default value for this property is False.

When set to TRUE, addresses that cannot be verified for purposes of DPV are returned unchanged with casing applied, that is, address correction rolls back to the original address. The error code indicates why the address is not verified.

In addition, when set to True, the setting of the DPVFailureAsError is ignored. However, if the original address includes a ZIP+4, the returned output will not include the +4 part of the ZIP Code.

## RuralRouteFormat

### Syntax

```
RuralRouteFormat  
  
long RuralRouteFormat(long *pVal) /* retrieving */  
long RuralRouteFormat(long newVal) /* setting */
```

### Description

Determines the format of returned rural route addresses.

### Data Type

Long

long

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is also the default value).

- 1 – Return standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

See also

[CheckAddress](#)

## SilentMode

Syntax

```
SilentMode  
  
long SilentMode(long *pVal) /* retrieving */  
long SilentMode(long newVal) /* setting */
```

Description

Indicates whether the ZIPTask process will display any dialog boxes.

Data Type

Long

long

Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

Return values

0 if False, -1 if True (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can read and set this property.

Set this property to True to disable all dialog boxes, including error messages.

The default value for this property is False.

## State

### Syntax

```
State  
  
long State(BSTR *pVal) /* retrieving */  
long State(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the state for this address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Call CheckAddress before attempting to retrieve the value of this property.

### See also

[CheckAddress](#)

## Street

### Syntax

```
Street  
  
long Street(BSTR *pVal) /* retrieving */  
long Street(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the street name of this address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

This property contains the name of the street for this address, such as the "Lake" in "416 Lake St."

### See also

[CheckAddress](#)

## Suffix

### Syntax

```
Suffix  
  
long Suffix(BSTR *pVal) /* retrieving */  
long Suffix(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the street suffix for this address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

A street suffix is the street type, often abbreviated, that follows the street name, such as the "St" in "416 Lake St."

### See also

[CheckAddress](#)

[SuffixFormat](#)

## SuffixFormat

### Syntax

```
SuffixFormat  
  
long SuffixFormat(long *pVal) /* retrieving */  
long SuffixFormat(long newVal) /* setting */
```

### Description

Determines how the returned street suffix is formatted.

### Data Type

Long

long

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is also the default value).

- 1 – Return standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

See also

[CheckAddress](#)

[Suffix](#)

## SuiteLinkFootnote

Syntax

```
SuiteLinkFootnote  
long SuiteLinkFootnote (BSTR *pVal)
```

Description

Indicates the results of a Suite<sup>Link</sup> lookup.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

This property returns a code indicating specific information about a Suite<sup>Link</sup> lookup, as described in the following table.

Value	Description
""	Was not processed by the Suite <sup>Link</sup> engine: the address did not qualify for a lookup within the Suite <sup>Link</sup> file. Only default high-rise addresses qualify for a Suite <sup>Link</sup> lookup.
A	The address was processed and secondary information was added to the resulting address.
00	The address was processed through the Suite <sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.

See also

[CheckAddress](#)

[UseSuiteLink](#)

## UnitDesignator

Syntax

```
UnitDesignator
```

```
long UnitDesignator(BSTR *pVal) /* retrieving */
```

```
long UnitDesignator(BSTR newVal) /* setting */
```

Description

Sets and retrieves the unit type for this address.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

A unit designator is the unit type, often abbreviated, that precedes the unit number, such as the "Ste" in "1301 5nd Ave Ste 2200."

## See also

[CheckAddress](#)

## UnitDesignatorFormat

### Syntax

```
UnitDesignatorFormat  
  
long UnitDesignatorFormat(long *pVal) /* retrieving */  
long UnitDesignatorFormat(long newVal) /* setting */
```

### Description

Determines the format of the returned unit designator of this address.

### Data Type

Long

long

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format; this is also the default value).
- 1 – Return standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

## See also

[CheckAddress](#)

[UnitDesignator](#)

## UnitNumber

### Syntax

```
UnitNumber
```

```
long UnitNumber(BSTR *pVal) /* retrieving */
```

```
long UnitNumber(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the unit number for this address.

### Data Type

String

BSTR

## Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

Call CheckAddress before attempting to retrieve the value of this property.

A unit number is the apartment or suite number in a building with multiple occupants but a single street address, such as the "2200" in "1301 5nd Ave Ste 2200."

## See also

[CheckAddress](#)

## UnitOutput

### Syntax

```
UnitOutput
```

```
long UnitOutput(long *pVal) /* retrieving */
```

```
long UnitOutput(long newVal) /* setting */
```

### Description

Determines on which address line returned unit information is placed.

### Data Type

Long

long

## Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPTask object.

Possible values are:

- 0 – Return at the end of the primary address line (default value).
- 1 – Return on the secondary address line (if secondary address is blank).
- 2 – Always return on the secondary address line (existing secondary address data will be overwritten).

## See also

[CheckAddress](#)

[AddressLine1](#)

[AddressLine2](#)

[UnitDesignator](#)

[UnitNumber](#)

## UpdateUncorrectedCityStZip

### Syntax

```
UpdateUncorrectedCityStZip  
  
long UpdateUncorrectedCityStZip(long *pVal) /* retrieving */  
long UpdateUncorrectedCityStZip(long newVal) /* setting */
```

### Description

Determines if the city, state and ZIP Code information should be returned for an address that could not be corrected.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

If property value is True, then city, state and ZIP Code data will be returned, when possible, for addresses that could not be corrected when CheckAddress is called.

The default value for this property is False.

See also

[CheckAddress](#)

[City](#)

[CityStateZip](#)

[State](#)

[ZipCode](#)

## Urbanization

Syntax

```
Urbanization  
  
long Urbanization(BSTR *pVal) /* retrieving */  
long Urbanization(BSTR newVal) /* setting */
```

Description

Sets and retrieves the urbanization data for this address.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

Call `CheckAddress` before attempting to retrieve the value of this property.

Urbanization data applies to Puerto Rico addresses only.

## See also

[CheckAddress](#)

## UseDPV

### Syntax

```
UseDPV  
  
long UseDPV(long *pVal) /* retrieving */  
long UseDPV(long newVal) /* setting */
```

### Description

Determines if this address should be validated with DPV.

### Data Type

Long

long

### Parameters

None

`pVal` – Returns 0 if False, -1 if True.

`newVal` – Set to 0 for False, non-zero value for True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

If property value is True, the address is checked with DPV (verifies deliverability) when CheckAddress is called.

DPV is now an integral and essential component of address correction as required by the USPS.

The default value for this property is True.

See also

[CheckAddress](#)

[DPVCoded](#)

[DPVFailureAsError](#)

[DPVFootnotes](#)

[DPVIndicator](#)

[DPVIsCMRA](#)

## UseGeocode

### Syntax

```
UseGeocode
```

```
long UseGeocode(long *pVal) /* retrieving */
```

```
long UseGeocode(long newVal) /* setting */
```

### Description

Determines if the ZIPTask should lookup geocode information for this address.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

## Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

If property value is True, geocode data will be returned when CheckAddress is called.

Geocode is a BCC Architect add-on.

## See also

[CheckAddress](#)

[CensusBlock](#)

[CensusTract](#)

[GeocodeFootnote](#)

[Latitude](#)

[Longitude](#)

[MSACode](#)

## UseLACS

### Syntax

```
UseLACS  
  
long UseLACS(long *pVal) /* retrieving */  
long UseLACS(long newVal) /* setting */
```

### Description

Determines if ZIPTask should lookup LACS<sup>Link</sup> information for this address.

### Data Type

Long

long

#### Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

#### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

If property value is True, ZIPTask will compare this address against the LACS<sup>Link</sup> database when CheckAddress is called.

The USPS requires LACS<sup>Link</sup> matching as part of CASS certified address correction.

#### See also

[CheckAddress](#)

[LACSFootnote](#)

[LACSIndicator](#)

## UseSuiteLink

#### Syntax

```
UseSuiteLink
```

```
long UseSuiteLink(long *pVal) /* retrieving */
```

```
long UseSuiteLink(long newVal) /* setting */
```

#### Description

Determines if ZIPTask should match this address to the Suite<sup>Link</sup> database.

## Data Type

Long

long

## Parameters

None

pVal – Returns 0 if False, -1 if True.

newVal – Set to 0 for False, non-zero value for True.

## Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

If property value is True, Suite<sup>Link</sup> will attempt to add missing secondary information when CheckAddress is called.

The default value for this property is True in batch processing, and False in single implementations.

## See also

[CheckAddress](#)

[SuiteLinkFootnote](#)

## ZipCode

### Syntax

ZipCode

```
long ZipCode(BSTR *pVal) /* retrieving */
```

```
long ZipCode(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the ZIP Code of this address.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

Call CheckAddress before attempting to retrieve the value of this property.

After a successful call to CheckAddress, the ZipCode property includes the +4 code.

### See also

[CheckAddress](#)

## ZipCoded

### Syntax

```
ZipCoded
```

```
long ZipCoded(long *pVal) /* retrieving */
```

### Description

Indicates that an address was coded with only a 5-digit ZIP Code.

### Data Type

Long

long

### Parameters

None

pVal – Returns 0 if False, -1 if True.

### Return values

0 if False, -1 if True. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckAddress before attempting to retrieve the value of this property.

### See also

[CheckAddress](#)

[Plus4Coded](#)

[ZipCode](#)

## The COM CASSTask Object for Correcting Batch Addresses

The BCC Architect CASSTask object performs address correction with the CASS certified address correction engine on a batch of addresses. CASSTask also standardizes abbreviations.

CASSTask provides a flexible interface through which you can control the amount of information returned for each address as well as the number of records returned during each batch process. In addition, CASSTask can also be set to indicate if a record was processed successfully on a previous occasion (see FLD\_SKIPPED\_CERTIFY in the MRTKTaskLib Field Names table). By checking this flag, a program can avoid processing previously corrected addresses, potentially speeding up processing.

As with all the BCC Architect objects, CASSTask should be created through the MRTKObjFactory.

### CASSTask Functions

The CASSTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

## PrepareTask

### Syntax

```
PrepareTask  
  
long PrepareTask()
```

### Description

Initializes and prepares the CASSTask object.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

This function must be called before any of the other functions or properties of CASSTask.

If this function is not called first, all subsequent CASSTask function calls will fail.

The one exception to this rule is calling SetProperty to set the path of the Address.cas file, which must be defined prior to calling PrepareTask. It is preferable, however, to set the data file path using the MRTKObjFactory.DataFilePath property.

### See also

[COM Factory Object](#)

## GetProperty

### Syntax

```
GetProperty(mrtkPropertyID)  
  
long GetProperty(long mrtkPropertyID, VARIANT *pVal)
```

### Description

Retrieves the value of a CASSTask property.

### Parameters

*mrtkPropertyID as Long*

The property ID of the property to get.

*mrtkPropertyID*

The property ID of the property to get.

*pVal*

Returns the value of the property.

#### Return values

The value of the property (as Variant). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

See the [CASSTask Properties](#) table for a list of property IDs.

## SetProperty

#### Syntax

```
 SetProperty(mrtkPropertyID, value)  
long SetProperty(long mrtkPropertyID, VARIANT value)
```

#### Description

Sets the value of a CASSTask property.

#### Parameters

*mrtkPropertyID as Long*

The property ID of the property to set.

*value as Variant*

The value of the property to set.

*mrtkPropertyID*

The property ID of the property to set.

*value*

The value of the property to set.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

See the [CASSTask Properties](#) table for a list of property IDs.

### ShowCASSWizard

#### Syntax

```
ShowCASSWizard
```

```
long ShowCASSWizard()
```

#### Description

Displays the CASS Agent Wizard, which walks the user through the correction process.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Call this function to display the pre-designed CASS Agent Wizard.

The CASS Agent Wizard provides a graphical interface that leads a user through the steps necessary to process a mailing list.

### ValidateProperties

#### Syntax

```
ValidateProperties
```

```
long ValidateProperties()
```

#### Description

Verifies that the task is set up correctly and ready to run.

## Parameters

None.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function verifies that the address matching engine is loaded and able to run and the input field list consists of the minimum set of fields, namely, Address Line 1 and either City/State or ZIP Code.

This function needs to be called before you call Update.

## See also

See the [CASSTask Properties](#) table for a list of property IDs.

## GetPropertySummary

### Syntax

```
String GetPropertySummary(mrtkPropertyID)
```

```
long GetPropertySummary(long mrtkPropertyID, BSTR *pVal)
```

### Description

Returns a description string for a property.

### Parameters

*MrtkPropertyID as Long*

The ID of the property for which you want summary information.

*mrtkPropertyID*

The ID of the property for which you want summary information.

*pVal*

Returns a BSTR that contains a description of the property specified by mrtkPropertyID.

## Return values

A description of the property specified by `mrtkPropertyID` (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function can be used to display the description of a property on screen to the user.

See the [CASSTask Properties](#) table for a list of property IDs.

## Update

### Syntax

```
Update (bstrAddressBlock)
```

```
long Update (BSTR *pbstrAddressBlock)
```

### Description

Process all the addresses contained in the record block.

### Parameters

#### *bstrAddressBlock as String*

In:

A string that contains `mrtkINPUT_BLOCK_RECORD_COUNT` addresses separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`.

Out:

Returns a block of corrected addresses that contains `mrtkINPUT_BLOCK_RECORD_COUNT` addresses separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`

#### *pbstrAddressBlock*

In:

A string that contains `mrtkINPUT_BLOCK_RECORD_COUNT` addresses separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`

Out:

Returns a block of corrected addresses that contains `mrtkINPUT_BLOCK_RECORD_COUNT` addresses separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You may want to experiment with the `mrtkINPUT_BLOCK_RECORD_COUNT` property. In preliminary tests, we have found the optimal setting to be around 25–50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500–1000. A setting of 25–50 makes the process run very slowly in client-server mode. The number of fields that you want returned (see `mrtkFIELD_LIST_OUT` property) greatly affects this number.

For optimal performance, only ask for fields that you want. The extra information requires additional lookups that slow processing.

If your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix), use the `UpdateV` function instead.

### See also

- [UpdateV](#)

See the [MRTK Global Properties](#) table for definition of:

- `mrtkINPUT_BLOCK_RECORD_COUNT`
- `mrtkDELIMITER_FIELD`
- `mrtkDELIMITER_RECORD`
- `mrtkFIELD_LIST_IN`
- `mrtkFIELD_LIST_OUT`

See the [COM Field Names](#) table for definition of:

- `FLD_SKIPPED_CERTIFY`

## UpdateV

### Syntax

```
UpdateV(bstrInBlock)
```

```
long UpdateV(BSTR bstrInBlock, BSTR *pbstrOutBlock)
```

### Description

Process the addresses contained in the record block.

### Parameters

#### *bstrAddressBlock as String*

In:

A string that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

Out:

Returns a block of corrected addresses that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD

#### *pbstrAddressBlock*

In:

A string that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD

Out:

Returns a block of corrected addresses that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You may want to experiment with the mrtkINPUT\_BLOCK\_RECORD\_COUNT property. In preliminary tests, we have found the optimal setting to be around 25-50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500-1000. A setting of 25-50 makes the process run very slowly in client-server mode..

The number of fields that you want returned (see `mrtkFIELD_LIST_OUT` property) greatly affects this number.

For optimal performance, only ask for fields that you want. The extra information requires additional lookups that slow processing.

You must use this function if your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix). Otherwise, you can use either this function or `Update`.

See also

- [Update](#)

See the [MRTK Global Properties](#) table for definition of:

- `mrtkINPUT_BLOCK_RECORD_COUNT`
- `mrtkDELIMITER_FIELD`
- `mrtkDELIMITER_RECORD`
- `mrtkFIELD_LIST_IN`
- `mrtkFIELD_LIST_OUT`

See the [COM Field Names](#) table for definition of:

- `FLD_SKIPPED_CERTIFY`

## ReviewErrors

### Syntax

```
ReviewErrors
```

```
long ReviewErrors()
```

### Description

Displays the Review Uncorrected Records dialog box.

## Parameters

None.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

When the Review Uncorrected Records dialog box is displayed, any addresses that are kept will be returned by the RetrieveReviewed function.

Before you can successfully call this function, set the ctREVIEW\_ERRORS property to True.

## See also

See the [CASSTask Properties](#) table for definition of:

- ctREVIEW\_ERRORS

## RetrieveReviewed

### Syntax

```
RetrieveReviewed(bstrAddressBlock)  
  
long RetrieveReviewed(BSTR *pbstrAddressBlock)
```

### Description

Retrieve the addresses that the user saved in the Review Uncorrected Records dialog box.

### Parameters

*bstrAddressBlock*

Returns a block of corrected addresses that contains etRECORD\_COUNT\_PER\_RECEIVE addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

*pbstrAddressBlock*

Returns a block of corrected addresses that contains etRECORD\_COUNT\_PER\_RECEIVE addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Before you can successfully call this function, you must set the ctREVIEW\_ERRORS property to True.

Only those addresses the user chooses to keep in the Review Uncorrected Records dialog box will be returned by this function.

If your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix), use the RetrieveReviewedV function instead.

#### See also

- [RetrieveReviewedV](#)

See the [MRTK Global Properties](#) table for definition of:

- etRECORD\_COUNT\_PER\_RECEIVE
- mrtkDELIMITER\_FIELD
- mrtkDELIMITER\_RECORD
- mrtkFIELD\_LIST\_OUT

See the [CASSTask Properties](#) table for definition of:

- ctREVIEW\_ERRORS

## RetrieveReviewedV

#### Syntax

```
RetrieveReviewedV  
  
long RetrieveReviewedV(BSTR *pbstrAddressBlock)
```

#### Description

Retrieve the addresses that the user saved in the Review Uncorrected Records dialog box.

#### Parameters

None

### *pbstrAddressBlock* –

Returns a block of corrected addresses that contains `etRECORD_COUNT_PER_RECEIVE` addresses that are separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`.

#### Return values

A block of corrected addresses that contains `etRECORD_COUNT_PER_RECEIVE` addresses that are separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD` (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Before you can successfully call this function, you must set the `ctREVIEW_ERRORS` property to True.

Only those addresses the user chooses to keep in the Review Uncorrected Records dialog box will be returned by this function.

You must use this function if your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix). Otherwise, you can use either this function or `RetrieveReviewed`.

#### See also

- [RetrieveReviewed](#)

See the [MRTK Global Properties](#) table for definition of:

- `etRECORD_COUNT_PER_RECEIVE`
- `mrtkDELIMITER_FIELD`
- `mrtkDELIMITER_RECORD`
- `mrtkFIELD_LIST_OUT`

See the [CASSTask Properties](#) table for definition of:

- `ctREVIEW_ERRORS`

## Print3553

### Syntax

```
Print3553(bstrPrinterName, bShowPrintSetupDlg)  
  
long Print3553(BSTR bstrPrinterName, long bShowPrintSetupDlg)
```

### Description

Prints the Address Correction Summary report (PS 3553).

### Parameters

#### *bstrPrinterName as String*

The name of the printer to which to print the report.

#### *bShowPrintSetupDlg as Long*

1 to show the Printer Setup dialog box.

#### *bstrPrinterName*

The name of the printer to which to print the report.

#### *bShowPrintSetupDlg–*

1 to show the Printer Setup dialog box.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

If bstrPrinterName is set to an empty string (""), then the default printer will be used to print the form.

If bShowPrintSetupDlg is set to 1, then the value for bstrPrinterName will be ignored.

### See also

- [PrintPreview3553](#)

## PrintPreview3553

### Syntax

```
PrintPreview3553  
  
long PrintPreview3553()
```

### Description

Previews the Address Correction Summary Report (PS 3553).

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#)

### See also

- [Print3553](#)

## SaveReportAsPDF

### Syntax

```
SaveReportAsPDF(MRTKReportID, bstrFileName, nFileOption)  
  
long SaveReportAsPDF(long MRTKReportID, BSTR bstrFileName, long  
nFileOption, BSTR *pbstrOutputFileName)
```

### Description

Saves a report as a PDF file.

### Parameters

*MRTKReportID as Long*

*bstrFileName as String*

The name of the file to save.

*nFileOption as Long*

Determines what the program does if the file already exists. Takes the following values:

- 0 – Overwrite if file with file name specified in `bstrFileName` already exists
- 1 – Prompt if file with file name specified in `bstrFileName` already exists
- 2 – Create new file if file with name specified in `bstrFileName` already exists; a number will be appended to `bstrFileName`
- 3 – Append if file with name specified in `bstrFileName` already exists (not yet implemented)

### *MRTKReportID*

The ID number of the report to save. Currently, this parameter accepts the following:

- 340 – CASS summary report (PS 3553)

### *bstrFileName*

The name of the file to save.

### *nFileOption*

Determines what the program does if the file already exists. Takes the following values:

- 0 – Overwrite if file with file name specified in `bstrFileName` already exists
- 1 – Prompt if file with file name specified in `bstrFileName` already exists
- 2 – Create new file if file with name specified in `bstrFileName` already exists; a number will be appended to `bstrFileName`
- 3 – Append if file with name specified in `bstrFileName` already exists (not yet implemented)

### *pbstrOutputFileName*

Returns the name of the saved file.

### Return values

The name of the saved file. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

Currently, PS 3553 is the only available report.

## See also

- [SaveReportsAsPDF](#)

## SaveReportsAsPDF

### Syntax

```
SaveReportsAsPDF(bstrFileName, nFileOption)
```

```
long SaveReportsAsPDF(BSTR bstrFileName, long nFileOption, BSTR *pb-  
strOutputFileName)
```

### Description

Save all address correction reports as a PDF file.

### Parameters

*bstrFileName* as String –

The name of the file to save.

*nFileOption* as Long –

Determines what the program does if the file already exists. Takes the following values:

- 0 – Overwrite if file with file name specified in *bstrFileName* already exists
- 1 – Prompt if file with file name specified in *bstrFileName* already exists
- 2 – Create new file if file with name specified in *bstrFileName* already exists; a number will be appended to *bstrFileName*
- 3 – Append if file with name specified in *bstrFileName* already exists (not yet implemented)

*bstrFileName* –

The name of the file to save.

*nFileOption* –

Determines what the program does if the file already exists. Takes the following values:

- 0 – Overwrite if file with file name specified in `bstrFileName` already exists
- 1 – Prompt if file with file name specified in `bstrFileName` already exists
- 2 – Create new file if file with name specified in `bstrFileName` already exists; a number will be appended to `bstrFileName`
- 3 – Append if file with name specified in `bstrFileName` already exists (not yet implemented)

#### *pbstrOutputFileName* –

Returns the name of the saved file.

#### Return values

The name of the saved file. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Currently, PS 3553 is the only available report.

#### See also

- [SaveReportAsPDF](#)

## AbortTask

### Syntax

```
AbortTask  
  
long AbortTask()
```

### Description

Ends the task before it finishes.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

None.

#### EndTask

#### Syntax

```
EndTask
```

```
long EndTask()
```

#### Description

Cleans up and releases any resources that the CASSTask was using.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Call this function when you are done with the CASSTask object.

This function will clear all data stored in the CASSTask, so make sure to retrieve and store that information if you wish to keep it.

## CASSTask Properties

The CASSTask properties shown below are members of the MRTKTASKLib.CASSTaskPropertyID enumeration. These enum names are used as arguments for the GetProperty and SetProperty functions. If you have added the BCC Architect files, then you can view all of the available enums in the Object Browser.

### ctABBREVIATE\_ADDRESS\_LINE

#### Enum Value

413

#### Data Type

BOOL

#### Description

If set to TRUE, CASSTask will return the primary address line with no more than 30 characters.

Address lines with less than 30 characters will not be abbreviated.

The default value is FALSE.

### ctADDRESS\_INPUT\_PREFERENCE

#### Enum Value

284

#### Data Type

Long

#### Description

This property determines which address to give preference to if both a street address and a PO Box address are found and matched.

This property only applies if the two addresses are input as separate address lines.

If addresses are input on the same address line, then USPS rules dictate that the PO Box address has priority, regardless of position.

Possible values are:

- 0 – By position, giving preference to the bottom address when there are two address lines. This is the default value.
- 1 – Prefer PO Box address.
- 2 – Prefer street address.

### ctALL\_CAPS

#### Enum Value

250

#### Data Type

BOOL

#### Description

Determines whether the corrected addresses will be returned in all capital letters.

This property has been superseded by the more flexible ctCASING property, but has been retained for backward compatibility.

### ctCASING

#### Enum Value

295

#### Data Type

Long

#### Description

Determines casing of returned address elements as follows:

- 0 – Upper case
- 1 – Lower case
- 2 – Mixed case (default value)

### ctCASS\_WIZARD\_CAPTION

#### Enum Value

264

#### Data Type

String

#### Description

This sets the title at the top of the CASS Wizard.

The default value is "CASS Wizard".

## ctCERTIFY\_FLAG

### Enum Value

251

### Data Type

Long

### Description

Indicates whether to skip records that have been previously corrected based on the CASSDate field.

Possible values include:

- 0 – Check only records not corrected with this issue. This is the default value.
- 1 – Check every record.
- 4 – Corrected with this issue. This value indicates which records to include when building the PS Form 3553 based on the CASSDate field.

All records passed in will be checked to see if they qualify to be added to the totals for the PS Form 3553.

**NOTE** If a record is coded but does not qualify, it will show up in the errors section of the report.

## ctCITY\_FORMAT

### Enum Value

294

### Data Type

Long

### Description

Determines the format of a processed city name:

- 0 – Return full city name. This is the default value.
- 1 – Return abbreviated city name (if one exists) of no more than 13 characters
- 2 – Return as input. THIS VALUE IS NO LONGER SUPPORTED.

## ctDPV\_FAILURE\_AS\_ERROR

### Enum Value

273

### Data Type

BOOL

### Description

Determines how the CASSTask handles an address that has unit information that does not validate with DPV.

The default value is FALSE.

Set to TRUE to treat addresses that have a valid primary, but fail DPV because of missing or invalid unit information, as uncorrected addresses. The ZIP + 4 Code will not be returned for these records.

Addresses only need to pass primary DPV processing to qualify as address corrected.

## ctDPV\_RESOLVE\_MULTIPLE\_RESPONSE

### Enum Value

408

### Data Type

Long

### Description

THIS PROPERTY IS OBSOLETE.

Determines if DPV is used to help resolve multiple responses when address matching.

Set to 1 for True, 0 for False.

The default value is 0.

## ctFIRM\_OUTPUT

### Enum Value

288

### Data Type

Long

### Description

Determines how the business name is handled if it is not input as FLD\_BUSINESS and is recognized as a firm name.

Possible values are:

- 0 – Return as input. This is the default value.
- 1 – Return as BusinessName (only if input BusinessName was blank)
- 2 – Always return as BusinessName (will overwrite input FLD\_BUSINESS)

## ctHIDE\_PROGRESS\_AFTER\_BATCH

### Enum Value

262

### Data Type

BOOL

### Description

If set to TRUE, no dialog box will be shown to the user after batch processing completes.

The default value is FALSE.

If the progress dialog box is displayed, the user can print the PS Form 3553 from this dialog box.

## ctHIGHWAYCONTRACT\_FORMAT

### Enum Value

289

#### Data Type

Long

#### Description

Determines format of returned highway contract addresses. Possible values are:

- 0 – Return standardized USPS abbreviation (USPS preferred format). This is the default value.
- 1 – Return standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

### ctKEEP\_ALIAS\_ADDRESS

#### Enum Value

296

#### Data Type

BOOL

#### Description

Determines how street name input as alias should be returned.

If property value is TRUE, the input street name alias is returned.

The default value, FALSE, returns the official street name from the USPS database.

### ctKEEP\_EXTRA\_PRIMARY\_DATA

#### Enum Value

297

#### Data Type

BOOL

#### Description

Determines if extra information in the primary address line is kept.

The default value is FALSE.

## ctLIST\_NAME

### Enum Value

278

### Data Type

String

### Description

The name of the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctLIST\_PROCESSOR

### Enum Value

269

### Data Type

String

### Description

The name of the person processing the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctMAILERS\_ADDRESS

### Enum Value

280

### Data Type

String

### Description

The address of the person processing the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctMAILERS\_CITY

### Enum Value

281

### Data Type

String

### Description

The city of the person processing the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctMAILERS\_NAME

### Enum Value

279

### Data Type

String

### Description

The name of the person processing the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctMAILERS\_STATE

### Enum Value

282

### Data Type

String

### Description

The state for the person processing the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctMAILERS\_ZIP

### Enum Value

283

### Data Type

String

### Description

The ZIP Code for the person processing the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

## ctMRTK\_VERSION

### Syntax

```
PTMRTK_VERSION
```

```
long PTMRTK_VERSION(BSTR *pVal)
```

### Description

Returns the current BCC Architect server version.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value.

### Notes

You can only read this property.

## ctPMB\_OUTPUT

### Enum Value

287

### Data Type

Long

### Description

Determines location of returned private mailbox number:

- 0 – Return on same line as unit information.
- 1 – Return as input. This is the default.

## ctPOBOX\_FORMAT

### Enum Value

299

### Data Type

Long

### Description

Determines format of returned PO Box addresses:

- 0 – Return with standardized USPS abbreviation. This is the USPS preferred format and default value.
- 1 – Return with standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

## ctPOSTDIRECTIONAL\_FORMAT

### Enum Value

292

### Data Type

Long

## Description

Determines the format of the returned postdirectional of an address:

- 0 – Return with standardized USPS abbreviation. This is the USPS preferred format and default value.
- 1 – Return with standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

## ctPREDIRECTIONAL\_FORMAT

### Enum Value

290

### Data Type

Long

## Description

Determines the format of the returned predirectional of an address:

- 0 – Return with standardized USPS abbreviation. This is the USPS preferred format and default value.
- 1 – Return with standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

## ctPRIMARY\_ADDRESS\_LENGTH\_LIMIT

### Enum Value

7278

### Data Type

Integer

## Description

**IMPORTANT** The ctPRIMARY\_ADDRESS\_LENGTH\_LIMIT property is designed to facilitate USPS CASS-certification testing and is not meant for general use. We strongly recommend that

you use other abbreviation properties to configure primary address line abbreviation behavior.

Sets the maximum character length of the primary address line (address line 1). Abbreviation of different parts of the address line is determined by the settings of other abbreviation properties.

It is not necessary to adjust the default setting unless you want to enforce that the abbreviation behavior use USPS street alias names where they exist. In that case, set this property to 31.

#### Valid Values

Default: 1000 (no limit)

31 (abbreviate to street name alias)

#### ctPRIMARY\_ADDRESS\_OUTPUT

##### Enum Value

285

##### Data Type

Long

##### Description

Determines the location of the returned primary address:

- 0 – Return as AddressLine1 if AddressLine2 is blank. Otherwise, the primary address line will be returned as AddressLine2 with the secondary address data in AddressLine1. This is the default value.
- 1 – Return as AddressLine2.

#### ctRECORD\_COUNT

##### Enum Value

258

##### Data Type

Long

##### Description

Sets and retrieves the total number of records to be processed.

## ctRETURN\_INPUT\_ADDRESS\_ON\_UNCONFIRMED\_DPV

### Enum Value

7255

### Data Type

Boolean

### Description

Determines whether to roll back corrected addresses that are not DPV confirmable.

When set to TRUE, addresses that cannot be verified for purposes of DPV are returned unchanged with casing applied, that is, address correction rolls back to the original address. The error code indicates why the address is not verified.

In addition, when set to TRUE, the setting of the ct\_DPV\_FAILURE\_AS\_ERROR is ignored. However, if the original address includes a ZIP+4, the returned output will not include the +4 part of the ZIP Code.

Read-only. The default value is FALSE.

## ctREVIEW\_ERRORS

### Enum Value

259

### Data Type

BOOL

### Description

Determines whether to display the Review Errors dialog box after batch processing completes.

The default value is TRUE.

## ctRURALROUTE\_FORMAT

### Enum Value

274

### Data Type

Long

## Description

Determines the format of returned rural route addresses:

- 0 – Return with standardized USPS abbreviation. This is the USPS preferred format and default value.
- 1 – Return with standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

## ctSHOW\_PROGRESS

### Enum Value

261

### Data Type

BOOL

### Description

Determines whether to show the Progress screen during batch processing.

The default value is TRUE.

## ctSILENT\_MODE

### Enum Value

206

### Data Type

BOOL

### Description

Determines whether to run the CASSTask in silent mode, which disables all dialogs boxes, including error messages.

The default value is FALSE.

## ctSUFFIX\_FORMAT

### Enum Value

291

## Data Type

Long

## Description

Determines the format of the returned suffix of an address:

- 0 – Return with standardized USPS abbreviation. This is the USPS preferred format and default value.
- 1 – Return with standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

**ctUNIT\_DESIGNATOR\_FORMAT**

## Enum Value

293

## Data Type

Long

## Description

Determines the format of the returned unit designator of an address:

- 0 – Return with standardized USPS abbreviation. This is the USPS preferred format and default value.
- 1 – Return with standardized USPS abbreviation with punctuation.
- 2 – Return full word(s).

**ctUNIT\_OUTPUT**

## Enum Value

286

## Data Type

Long

## Description

Determines the location of returned unit information:

- 0 – Return at the end of the primary address line.
- 1 – Return on the secondary address line if it is blank.
- 2 – Always return on secondary address line. Existing secondary address data will be overwritten.

## ctUPDATE\_UNCORRECTED\_CITYSTZIP

### Enum Value

298

### Data Type

BOOL

## Description

Determines if corrected city, state and ZIP Code information should be returned for uncorrected addresses, if possible.

The default value is FALSE.

## ctUPDATECASE\_BUSINESS

### Enum Value

400

### Data Type

BOOL

## Description

Determines if the casing as specified by ctCASING is applied to business name.

The default value is TRUE.

## ctUPDATECASE\_NAMES

### Enum Value

401

#### Data Type

BOOL

#### Description

Determines if the casing as specified by ctCASING is applied to name fields.

The default value is FALSE.

### ctUSE\_COUNTRY

#### Enum Value

7167

#### Data Type

Boolean

#### Description

Use this property to incorporate the use of the Country field in processing foreign addresses.

Relies on data in the field FLD\_COUNTRY.

The default value for this property is False.

### ctUSE\_SUITELINK

#### Enum Value

412

#### Data Type

BOOL

#### Description

This property is now obsolete. Suite<sup>Link</sup> is now a requirement of CASS processing.

COM CASSTask Properties Summary Table

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctABBREVIATE_ADDRESS_LINE	413	BOOL	FALSE	If set to TRUE, CASSTask will return the primary address line with no more than 30 characters. Address lines with less than 30 characters will not be abbreviated.
ctADDRESS_INPUT_PREFERENCE	284	Long	0	<p>This property determines which address to give preference to if both a street address and a PO Box address are found and matched. This property only comes into play if the two addresses are input as separate address lines. If they are input on the same address line, then USPS rules dictate that the PO Box address has priority, regardless of position. Possible values are:</p> <ul style="list-style-type: none"> <li>• 0 – By position, giving preference to the bottom address when there are two address lines</li> <li>• 1 – Prefer PO Box address</li> <li>• 2 – Prefer street address</li> </ul>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctALL_CAPS	250	BOOL	False	<ul style="list-style-type: none"> <li>• TRUE = Return all caps</li> <li>• FALSE = Return mixed case</li> </ul> <p><b>NOTE</b> This property has been superseded by the more flexible ctCASING property, but has been retained for backward compatibility.</p>
ctCASING	295	Long	2	<p>Determines casing of returned address elements as follows:</p> <ul style="list-style-type: none"> <li>• 0 – Upper case</li> <li>• 1 – Lower case</li> <li>• 2 – Mixed case</li> </ul>
ctCASS_WIZARD_CAPTION	264	String	"CASS Wizard"	<p>Caption that appears on CASS Wizard dialog box window.</p>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctCERTIFY_FLAG	251	Long	0	<p>Indicate whether to skip records that have been previously corrected based on the CASSDate field:</p> <ul style="list-style-type: none"> <li>• 0 – Check only records not corrected with this issue</li> <li>• 1 – Check every record</li> <li>• Indicate records to include when building the PS Form 3553 based on the CASSDate field:</li> <li>• 4 – Corrected with this issue</li> </ul> <p>All records passed in will be checked to see if they "qualify" to be added to the totals for the PS Form 3553.</p> <p><b>NOTE</b> If a record is coded but does not qualify, it will show up in the errors section of the report.</p>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctCITY_FORMAT	294	Long	0	<p>Determines format of returned city name:</p> <ul style="list-style-type: none"> <li>• 0 – Return full city name (default value)</li> <li>• 1 – Return abbreviated city name (if one exists) of no more than 13 characters</li> <li>• 2 – Return as input. THIS VALUE IS NO LONGER SUPPORTED.</li> </ul>
ctDPV_FAILURE_AS_ERROR	273	BOOL	FALSE	<p>Determines how an address that has unit information that does not confirm with DPV is handled. Set to TRUE to treat addresses that have a valid primary, but fail DPV because of missing or invalid unit information, as uncorrected addresses. The ZIP+4 code will not be returned for these records.</p>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctDPV_RESOLVE_MULTIPLE_RESPONSE	408	Long	0	<p>Determines if DPV is used to help resolve multiple responses when address matching. Set to 1 for True, 0 for False.</p> <p>THIS PROPERTY IS OBSOLETE.</p>
ctFIRM_OUTPUT	288	Long	0	<p>Determines how the business name is handled if it is not input as FLD_BUSINESS and is recognized as a firm name:</p> <ul style="list-style-type: none"> <li>• 0 – Return as input</li> <li>• 1 – Return as BusinessName (only if input BusinessName was blank)</li> <li>• 2 – Always return as BusinessName (will overwrite input FLD_BUSINESS)</li> </ul>
ctHIDE_PROGRESS_AFTER_BATCH	262	BOOL	FALSE	<p>TRUE to hide results dialog after the completion of batch processing. If the progress dialog is shown the user can print the PS Form 3553 from this dialog.</p>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctHIGHWAYCONTRACT_FORMAT	289	Long	0	<p>Determines format of returned highway contract addresses:</p> <ul style="list-style-type: none"> <li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li> <li>• 1 – Return standardized USPS abbreviation with punctuation</li> <li>• 2 – Return full word(s)</li> </ul>
ctKEEP_ALIAS_ADDRESS	296	BOOL	FALSE	<p>Determines how street name input as alias should be returned. If property value is True, the input street name alias is returned. Otherwise, the "official" street name from the USPS database is returned.</p>
ctKEEP_EXTRA_PRIMARY_DATA	297	BOOL	FALSE	<p>Determines if extra information in the primary address line is kept.</p>
ctLIST_NAME	278	String	EMPTY	<p>Name of list to be printed on PS Form 3553</p>
ctLIST_PROCESSOR	269	String	EMPTY	<p>Name of list processor to be printed on PS Form 3553.</p>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctMAILERS_ADDRESS	280	String	EMPTY	Address of mailer to be printed on PS Form 3553
ctMAILERS_CITY	281	String	EMPTY	City of mailer to be printed on PS Form 3553
ctMAILERS_NAME	279	String	EMPTY	Name of mailer to be printed on PS Form 3553
ctMAILERS_STATE	282	String	EMPTY	State of mailer to be printed on PS Form 3553
ctMAILERS_ZIP	283	String	EMPTY	ZIP code of mailer to be printed on PS Form 3553
ctMRTK_VERSION	212	String	EMPTY	Returns the current BCC Architect server version.
ctPMB_OUTPUT	287	Long	1	Determines location of returned private mailbox number: <ul style="list-style-type: none"> <li>• 0 – Return on same line as unit information</li> <li>• 1 – Return as input</li> </ul>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctPOBOX_FORMAT	299	Long	0	<p>Determines format of returned PO Box addresses:</p> <ul style="list-style-type: none"> <li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li> <li>• 1 – Return standardized USPS abbreviation with punctuation</li> <li>• 2 – Return full word(s)</li> </ul>
ctPOSTDIRECTIONAL_FORMAT	292	Long	0	<p>Determines format of returned postdirectional of an address:</p> <ul style="list-style-type: none"> <li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li> <li>• 1 – Return standardized USPS abbreviation with punctuation</li> <li>• 2 – Return full word(s)</li> </ul>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctPREDIRECTIONAL_FORMAT	290	Long	0	<p>Determines format of returned predirectional of an address:</p> <ul style="list-style-type: none"> <li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li> <li>• 1 – Return standardized USPS abbreviation with punctuation</li> <li>• 2 – Return full word(s)</li> </ul>
ctPRIMARY_ADDRESS_LENGTH_LIMIT	7278	Integer	1000	<p>Sets the maximum character length of the primary address line (address line 1). Abbreviation of different parts of the address line is determined by the settings of other abbreviation properties.</p> <p>It is not necessary to adjust the default setting unless you want to enforce that the abbreviation behavior use USPS street alias names where they exist. In that case, set this property to 31.</p>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctPRIMARY_ADDRESS_OUTPUT	285	Long	0	<p>Determines location of returned primary address:</p> <ul style="list-style-type: none"> <li>• 0 – Return as AddressLine1 (if AddressLine2 is blank; otherwise, the primary address line will be returned as AddressLine2 with the secondary address data in AddressLine1)</li> <li>• 1 – Return as AddressLine2</li> </ul>
ctRECORD_COUNT	258	Long	0	The total number of records to be processed
ctRETURN_INPUT_ADDRESS_ON_UNCONFIRMED_DPV	7255	BOOL	FALSE	Determines whether to roll back corrected addresses that are not DPV confirmable.
ctREVIEW_ERRORS	259	BOOL	TRUE	TRUE to display the Review Errors window after batch processing

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctRURALROUTE_FORMAT	274	Long	0	<p>Determines format of returned rural route addresses:</p> <ul style="list-style-type: none"> <li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li> <li>• 1 – Return standardized USPS abbreviation with punctuation</li> <li>• 2 – Return full word(s)</li> </ul>
ctSHOW_PROGRESS	261	BOOL	TRUE	TRUE to show progress dialog during CASS processing
ctSILENT_MODE	206	BOOL	FALSE	TRUE to run the task in silent mode, which disables all dialogs, including error messages

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctSUFFIX_FORMAT	291	Long	0	<p>Determines format of returned suffix of an address:</p> <ul style="list-style-type: none"><li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li><li>• 1 – Return standardized USPS abbreviation with punctuation</li><li>• 2 – Return full word(s)</li></ul>
ctUNIT_DESIGNATOR_FORMAT	293	Long	0	<p>Determines format of returned unit designator of an address:</p> <ul style="list-style-type: none"><li>• 0 – Return standardized USPS abbreviation (USPS preferred format)</li><li>• 1 – Return standardized USPS abbreviation with punctuation</li><li>• 2 – Return full word(s)</li></ul>

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctUNIT_OUTPUT	286	Long	0	<p>Determines location of returned unit information:</p> <ul style="list-style-type: none"> <li>• 0 – Return at end of primary address line</li> <li>• 1 – Return on secondary address line (if secondary address is blank)</li> <li>• 2 – Always return on secondary address line (existing secondary address data will be overwritten)</li> </ul>
ctUPDATE_UNCORRECTED_CITYSTZIP	298	BOOL	FALSE	Determines if city, state and ZIP Code should be returned for uncorrected addresses, when possible.
ctUPDATECASE_BUSINESS	400	BOOL	TRUE	Determines if selected casing option is applied to business name.
ctUPDATECASE_NAMES	401	BOOL	TRUE	Determines if selected casing option is applied to name fields.

CASSTask Properties	Enum Value	Data Type	Default Value	Description
ctUSE_COUNTRY	7167	BOOL	FALSE	Use this property to incorporate the use of the Country field in processing foreign addresses.  Relies on data in the field FLD_COUNTRY.
ctUSE_SUITE_LINK	412	BOOL	TRUE	Obsolete. Suite <sup>Link</sup> is now a requirement of CASS processing.

## The COM CASSReportTask Object for Creating USPS Postal Reports

**NOTE** PS Form 3553 can now be printed using the Print3553/PrintPreview3553 methods of the CASSTask object. We recommend that you use this approach. The CASSReportTask object has been retained for backward compatibility.

The CASSReportTask object prints an Address Correction report (PS Form 3553). CASSReportTask prints this report for the last batch of addresses processed using the CASSTask or PresortTask object. Therefore, if you wish to print a more current report, you should batch process a mailing list using the CASSTask or PresortTask object before calling the CASSReportTask object.

The CASSReportTask object should be created through the COM object factory MRTKObjFactory.

### CASSReportTask Functions

The CASSReportTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

#### PrepareTask

Syntax

```
PrepareTask
```

```
long PrepareTask()
```

## Description

Initialize and prepare the object.

## Parameters

None

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You must call this function before any of the other functions or properties of CASSReportTask. If you do not, any CASSReportTask functions calls will fail.

The only exception to this is calling the SetProperty function to set the path of the Address.cas file. We recommend, however, that you set the data file path using the MRTKObjFactory.DataFilePath property. This way, all BCC Architect objects will have the correct data file path.

## See also

- [COM Factory Object](#)

## GetProperty

### Syntax

```
GetProperty(mrtkPropertyID)
```

```
long GetProperty(long mrtkPropertyID, VARIANT *pVal)
```

### Description

Retrieves a CASSReportTask property value.

### Parameters

*MrtkPropertyID as Long*

The ID of the property to retrieve.

*MrtkPropertyID*

The ID of the property to retrieve.

### *pVal*

Returns the value of the property.

#### Return values

The value of the property (as Variant). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

#### See also

See the CASSReportTask Properties table for a list of property IDs.

### SetProperty

#### Syntax

```
SetProperty(mrtkPropertyID, value)  
long SetProperty(long mrtkPropertyID, VARIANT value)
```

#### Description

Sets a CASSReportTask property value.

#### Parameters

##### *mrtkPropertyID as Long*

The ID of the property to set.

##### *value as Variant*

The value of the property to set.

##### *mrtkPropertyID*

The ID of the property to set.

##### *value*

The value of the property to set.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

### See also

See the CASSReportTask Properties table for a list of property IDs.

## ValidateProperties

### Syntax

```
ValidateProperties  
long ValidateProperties()
```

### Description

Verify that the task is set up correctly and ready to run.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You must call this function before you call Print3553 or PrintPreview3553.

It is very important to call EndTask, after processing the last batch of addresses, before calling ValidateProperties.

### See also

See the CASSReportTask Properties table for a list of property IDs.

## Print3553

### Syntax

```
Print3553  
long Print3553()
```

### Description

Prints the Address Correction report (PS Form 3553).

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You must call `ValidateProperties` before you call this function.

#### See also

[ValidateProperties](#)

## PrintPreview3553

#### Syntax

```
PrintPreview3553  
long PrintPreview3553()
```

#### Description

Displays the Address Correction report (PS Form 3553) on your screen.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You must call `ValidateProperties` before you call this function.

#### See also

[ValidateProperties](#)

## EndTask

### Syntax

```
EndTask  
  
long EndTask()
```

### Description

Cleans up and releases any resources that the CASSReportTask was using.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Call this function when you are done with the CASSReportTask object.

This function will clear all data stored in the CASSReportTask, so make sure to retrieve and store that information if you wish to keep it.

## CASSReportTask Properties

The CASSReportTask properties shown below are members of the MRTKTASKLib.CASSReportTaskPropertyID enumeration. These enum names are used as arguments for the GetProperty and SetProperty functions. If you have added the BCC Architect files, then you can view all of the available enums in the Object Browser.

### crLIST\_NAME

#### Enum Value

278

#### Data Type

String

#### Description

The name of the mailing list, which will be printed on PS Form 3553.

The default value is an empty string.

#### crLIST\_PROCESSOR

##### Enum Value

269

##### Data Type

String

##### Description

The name of the person processing the list, which will be printed on PS Form 3553.

The default value is an empty string.

#### crMAILERS\_ADDRESS

##### Enum Value

280

##### Data Type

String

##### Description

The address of the company processing the list, which will be printed on PS Form 3553.

The default value is an empty string.

#### crMAILERS\_CITY

##### Enum Value

281

##### Data Type

String

##### Description

The city for the company processing the list, which will be printed on PS Form 3553.

The default value is an empty string.

## crMAILERS\_NAME

### Enum Value

279

### Data Type

String

### Description

The name of the company processing the list, which will be printed on PS Form 3553.

The default value is an empty string.

## crMAILERS\_STATE

### Enum Value

282

### Data Type

String

### Description

The state for the company processing the list, which will be printed on PS Form 3553.

The default value is an empty string.

## crMAILERS\_ZIP

### Enum Value

283

### Data Type

String

### Description

The ZIP Code for the company processing the list, which will be printed on PS Form 3553.

The default value is an empty string.

## crPRINTER\_NAME

### Enum Value

275

### Data Type

String

### Description

The name of printer on which to print the PS Form 3553.

If the printer is a network printer the name must be the full name (e.g., `\\Server\HP Laser Jet 5`).

To find the full printer name, print a test page from the print driver.

The default value is the name of the default printer.

## crSHOW\_PAGE\_SETUP

### Enum Value

276

### Data Type

BOOL

### Description

Not currently implemented.

## crSHOW\_PRINT\_DIALOG

### Enum Value

277

### Data Type

BOOL

### Description

Determines whether the CASSReport Task will show a standard Print dialog box before printing.

The default value is TRUE.

## COM MRTK Report IDs Summary Table

MRTKReportIDs	Enum Value	Description
prPRINT_POSTAGE_SUMMARY	656	Determines whether to print the Postage Summary report.
prREPORT_CASSFORM	340	Specifies printing the USPS Form 3553 (required for Automation presorts)
prREPORT_CONTAINER_REPORT	361	Determines whether to print the Container Listing report.
prREPORT_DROP_SHIPMENT	362	Determines whether to print the Plant-Verified Drop Shipment (PVDS) report (PS Form 8125).
prREPORT_FACING_SLIPS	7110	TRUE to print facing slips
prREPORT_FILE_MAILDAT	364	Specifies saving the Mail.dat files to the specified folder
prREPORT_FILE_TRAYLABELS	345	Specifies saving the tray/sack labels to a text file
prREPORT_MOVE_JOBS_IDS	363	
prREPORT_MAIL_SORT_SUMMARY	350	Determines whether to print the Mail Sort Summary report.
prREPORT_MANIFEST	652	Specifies printing the mixed-weights manifest report
prREPORT_MOVE_ACTIVATION	353	Specifies printing the NCOA <sup>Link</sup> account activation forms
prREPORT_MOVE_NDI	354	Specifies printing the NCOA <sup>Link</sup> summary report

MRTKReportIDs	Enum Value	Description
prREPORT_PALLET_LABELS	7109	TRUE to print pallet labels.
prREPORT_POSTAGE	337	Specifies printing the USPS postage form; the actual form printed depends on the presort chosen
prREPORT_PRESORT	336	Specifies printing the BCC Software Presort Report
prREPORT_QUALIFICATION	335	Specifies printing the USPS Qualification Report (required for presort discount)
prREPORT_SHIPPING_SERVICES_FILE	7135	<p>In/Out.</p> <p>Indicates whether to generate the Shipping Services file.</p> <ul style="list-style-type: none"> <li>• 0= false</li> <li>• 1=true</li> </ul> <p>The default value is 0.</p>
prREPORT_SHIPPING_SERVICES_FILE_PATH	7136	The Shipping Services file path and name.
prREPORT_TRAY_LABELS_ONLY	7123	TRUE to print tray labels only, without additional documents.
prREPORT_TRAYLABELS	338	Specifies printing the tray/sack labels (required for presort discount)
prREPORT_ZIPLISTING	355	Specifies printing the ZIP Code listing report
prREPORT_ZONE	339	Specifies printing the Zone report, which details the piece count sent to a particular out-of-county zone for a Periodicals mailing

## COM CASSReportTask Properties Summary Table

CASSReport Task Properties	Enum Value	Data Type	Default Value	Description
crLIST_NAME	278	String	EMPTY	Name of list to be printed on PS Form 3553
crLIST_PROCESSOR	269	String	EMPTY	Name of list processor to be printed on PS Form 3553
crMAILERS_ADDRESS	280	String	EMPTY	Address of mailer to be printed on PS Form 3553
crMAILERS_CITY	281	String	EMPTY	City of mailer to be printed on PS Form 3553
crMAILERS_NAME	279	String	EMPTY	Name of mailer to be printed on PS Form 3553
crMAILERS_STATE	282	String	EMPTY	State of mailer to be printed on PS Form 3553
crMAILERS_ZIP	283	String	EMPTY	ZIP code of mailer to be printed on PS Form 3553
crPRINTER_NAME	275	String	Default Printer	Name of printer to send the PS Form 3553 to.  <b>NOTE</b> If the printer is a network printer the name must be the full name (e.g., <code>\\Server\HP Laser Jet 5</code> ). To find the full printer name, print a test page from the print driver.
crSHOW_PAGE_SETUP	276	BOOL	FALSE	Not currently implemented.
crSHOW_PRINT_DIALOG	277	BOOL	TRUE	TRUE to show the standard print dialog

## COM ReportWizard Properties Summary Table

These properties can and should be used with PresortTask.

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwAGENT_ADDRESS_ADDR	511	String	EMPTY	Address of Agent
rwAGENT_ADDRESS_CITY	512	String	EMPTY	City of Agent
rwAGENT_ADDRESS_CONTACT	515	String	EMPTY	Contact name of Agent
rwAGENT_ADDRESS_EMAIL	516	String	EMPTY	E-mail of Agent
rwAGENT_ADDRESS_NAME	510	String	EMPTY	<p>The company name of the Agent preparing the mailing. An Agent can prepare a mailing on behalf of an organization.</p> <p><b>NOTE</b> This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.</p>
rwAGENT_ADDRESS_PHONE	517	String	EMPTY	Phone of Agent
rwAGENT_ADDRESS_STATE	513	String	EMPTY	State of Agent
rwAGENT_ADDRESS_ZIP	514	String	EMPTY	ZIP code of Agent
rwMAILING_DATE	528	String	date of pre-sort	The date of mailing
rwNEWSPAPER_LABELS	564	BOOL	FALSE	Not currently implemented

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwNON_ADVERTISE_AMOUNT	526	Long	0	The percentage of non-advertising content for a Periodicals mail piece
rwORG_ADDRESS_ADDR	521	String	EMPTY	Address of Organization
rwORG_ADDRESS_CITY	522	String	EMPTY	City of Organization
rwORG_ADDRESS_CONTACT	535	String	EMPTY	Contact name of Organization
rwORG_ADDRESS_EMAIL	536	String	EMPTY	E-mail of Organization
rwORG_ADDRESS_NAME	520	String	EMPTY	The company name of the organization on whose behalf a mailing is being prepared.  <div style="border-left: 2px solid #0070C0; padding-left: 10px; margin-left: 20px;"> <p><b>NOTE</b> This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.</p> </div>
rwORG_ADDRESS_PHONE	537	String	EMPTY	Phone of Organization
rwORG_ADDRESS_STATE	523	String	EMPTY	State of Organization
rwORG_ADDRESS_ZIP	524	String	EMPTY	ZIP code of Organization

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwORG_MAILER_ID	539	String	EMPTY	This property no longer has any effect. Use ptMAIL_OWNER_MAILER_ID or ptMAILING_AGENT_MAILER_ID instead.
rwPAYMENT_ACCOUNT_NUMBER	7178	String	EMPTY	Sets the Mail Anywhere account number or the EPS account number. Accepts up to 20 alphanumeric characters.
rwPAYMENT_OPTION	7240	String	0 for periodicals; otherwise 5	<p>A string that specifies the Mail.dat MPA postage payment option.</p> <ul style="list-style-type: none"> <li>0 – CPP</li> <li>1 – PVDS</li> <li>2 – EPS</li> <li>3 – Debit</li> <li>4 – Billing</li> <li>5 – Other</li> </ul> <p><b>NOTE</b> Option 2 was changed from CAPS to EPS in August – September 2019 Service Pack 1.</p>
rwPAYMENT_TYPE	531	Long	0	Not currently implemented

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwPERMIT_ADDRESS_ADDR	501	String	EMPTY	Address of permit holder
rwPERMIT_ADDRESS_CITY	502	String	EMPTY	City of permit holder
rwPERMIT_ADDRESS_CONTACT	505	String	EMPTY	Contact name of permit holder
rwPERMIT_ADDRESS_EMAIL	506	String	EMPTY	E-mail of permit holder
rwPERMIT_ADDRESS_NAME	500	String	EMPTY	The company name of the permit holder used to submit this mailing.
rwPERMIT_ADDRESS_STATE	503	String	EMPTY	State of permit holder
rwPERMIT_ADDRESS_ZIP	504	String	EMPTY	ZIP code of permit holder
rwPERMIT_CAPS_CUSTOMER_ID	6705	String	EMPTY	<p>Deprecated. ID for the Centralized Account Processing System (CAPS) electronic payment system.</p> <p>If used, the value in this field is combined with that in the rwPRINT_PRESORT_MAILER_ID field and can be used to identify a mailing on the <i>PostalOne!</i> dashboard.</p>

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwPERMIT_MAILER_ID	508	String	EMPTY	This property no longer supported. Use ptMAIL_OWNER_MAILER_ID or ptMAILING_AGENT_MAILER_ID instead.
rwPERMIT_NONPROFIT_AUTH_NO	507	String	EMPTY	Specifies the permit holder's nonprofit authorization number.
rwPERMIT_NUMBER	529	String	EMPTY	Permit number of permit holder
rwPOST_OFFICE	527	String	EMPTY	Post Office of mailing
rwPOSTAGE_AMOUNT	525	Long	EMPTY	The minimum postage affixed to each piece when the payment method is meter minimum or stamp
rwPRINT_PRESORT_CITYSTATE	542	String	EMPTY	Mailer's city and state to be printed on tray/sack labels
rwPRINT_PRESORT_MAILER_ID	544	String	EMPTY	<p>Deprecated. The Job ID to be printed on Qualification Report.</p> <p>If used, the value in this field is combined with that in the rwPERMIT_CAPS_CUSTOMER_ID field and can be used to identify a mailing on the <i>PostalOne!</i> dashboard.</p>

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwPRINT_PRESORT_NAME	540	String	EMPTY	Mailer's company name to be printed on tray/sack labels
rwPRINT_PRESORT_PUB_ID	543	String	EMPTY	Publication ID number to be printed on Qualification Report for a Periodicals mailing
rwPRINT_PRESORT_ZIPCODE	541	String	EMPTY	Mailer's ZIP code to be printed on tray/sack labels
rwSACK_LABEL_COLUMNS	561	Long	2	Number of sack labels across
rwSACK_LABEL_CONTINUOUS	557	BOOL	FALSE	TRUE if sack labels are printed on continuous labels (dot-matrix printer)
rwSACK_LABEL_HEIGHT	559	Float	1 in. or 0	Height of sack label in inches if printing to a sheet; not set if printing to continuous paper
rwSACK_LABEL_LEFT_MARGIN	563	Float	0.5 in.	Left margin in inches to start printing text on sack labels
rwSACK_LABEL_ROWS	560	Long	10 or 11	Number of rows of labels: 10 for sheet, 11 for continuous
rwSACK_LABEL_TOP_MARGIN	562	Float	0.5 in. or 0	Top margin in inches to start printing text; zero if printing to continuous paper

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwSACK_LABEL_WIDTH	558	Float	3.25 in.	Width of sack label in inches
rwTELEPHONE	530	String	EMPTY	Telephone number of the permit holder
rwTRAY_LABEL_COLUMNS	554	Long	2	Number of tray labels across
rwTRAY_LABEL_CONTINUOUS	550	BOOL	FALSE	TRUE if tray labels are printed on continuous labels (dot-matrix printer)
rwTRAY_LABEL_HEIGHT	552	Float	2 in. or 0	Height of tray label if printing to a sheet; not set if printing to continuous paper
rwTRAY_LABEL_HORIZ_PITCH		FLOAT	EMPTY	Determines the horizontal distance between the left edge of the left-most tray label and the left edge of the tray label to the right of it on the page
rwTRAY_LABEL_LEFT_MARGIN	556	Float	0.5 in.	Left margin in inches to start printing text on tray labels
rwTRAY_LABEL_PAGE_HEIGHT		FLOAT	EMPTY	Determines the vertical dimension of the tray label, in inches
rwTRAY_LABEL_PAGE_WIDTH		FLOAT	EMPTY	Determines the vertical dimension of the tray label, in inches

ReportWizard Properties	Enum Value	Data Type	Default Value	Description
rwTRAY_LABEL_ROWS	553	Long	5 or 7	Number of rows of labels: 5 for sheet, 7 for continuous
rwTRAY_LABEL_TOP_MARGIN	555	Float	0.5 In. or 0	Top margin in inches to start printing text; zero if printing to continuous paper
rwTRAY_LABEL_VERT_PITCH		FLOAT	0.5 or 0	Determines the vertical distance between the top edge of the first tray label and the top edge of the tray label below it on the page, in inches
rwTRAY_LABEL_WIDTH	551	Float	3.25 in.	Width of tray label in inches

## The COM MOVETask Object for Updating Moved Addresses

The MOVETask object processes a list of addresses and updates it with the latest National Change of Address (NCOA) data.

The process occurs over three phases.

1. The addresses are sent in blocks to the Move Update server, and the addresses are processed for address correction.
2. The list is compared against the NCOA database, and records that match a changed address are updated to the new address.
3. The list is retrieved, again in blocks. All processed addresses are returned with a results code describing the outcome of the NCOA<sup>Link</sup> processing.

After the job is finished, you can print the Address Correction report (PS Form 3553) and the NCOA<sup>Link</sup> Summary report. To review results, compare the report to the NCOA<sup>Link</sup> Results Codes Summary Table.

MOVETask provides a flexible interface through which you can control the amount of information returned for each address as well as the number of records that are sent or retrieved with each block. As with all the BCC Architect Task objects, MOVETask can be created through the MRTKObjFactory.

## MOVETask Functions

The MOVETask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

### PrepareTask

#### Syntax

```
PrepareTask  
  
long PrepareTask()
```

#### Description

Initialize and prepare the MOVETask object.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You must call this function before any of the other functions or properties of MOVETask. Failing to do so will cause subsequent function calls to fail.

The one exception is calling SetProperty with the mrtkMAILROOM\_SERVER\_LIST property to set the location of the Move Update Server. You must do this prior to calling PrepareTask.

#### See also

[COM Factory Object](#)

## GetProperty

### Syntax

```
GetProperty(mrtkPropertyID)  
  
long GetProperty(long mrtkPropertyID, VARIANT *pVal)
```

### Description

Retrieve a MOVETask property value.

### Parameters

*mrtkPropertyID as Long*

The ID of the property to retrieve.

*mrtkPropertyID*

The ID of the property to retrieve.

*pVal*

Returns the value of the property.

### Return values

The value of the property (as Variant). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can set and retrieve most of the CASSTask properties with this MOVETask function.

### See also

#### [SetProperty](#)

See the [CASSTask Properties](#), [MOVETask Properties](#), and [MRTK Global Properties](#) tables for a list of property IDs.

## SetProperty

### Syntax

```
SetProperty(mrtkPropertyID, value)  
  
long SetProperty(long mrtkPropertyID, VARIANT value)
```

## Description

Set a MOVETask property value.

## Parameters

*mrtkPropertyID as Long*

The ID of the property to set.

*value as Variant–*

The value of the property to set.

*mrtkPropertyID*

The ID of the property to set.

*value–*

The value of the property to set.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can set and retrieve most of the CASSTask properties with this MOVETask function.

## See also

- [GetProperty](#)
- [ShowMoveWizard](#)

See the [CASSTask Properties](#), [MOVETask Properties](#), and [MRTK Global Properties](#) tables for a list of property IDs.

## ShowMOVEWizard

### Syntax

```
ShowMOVEWizard
```

```
long ShowMOVEWizard()
```

### Description

Display the MOVE Agent Wizard.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Call this function to provide a graphical interface and lead the user through the steps necessary to configure and process a list with the NCOA<sup>Link</sup> service.

Properties set through the wizard do not have to be set through SetProperty.

### See also

[SetProperty](#)

## ValidateProperties

### Syntax

```
ValidateProperties  
long ValidateProperties()
```

### Description

Verify that the task is set up correctly and ready to run.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You must call this function after setting the property values but before you call Send and begin processing.

See also

### [SetProperty](#)

See the [CASSTask Properties](#), [MOVETask Properties](#), and [MRTK Global Properties](#) tables for a list of property IDs.

Send

Syntax

```
Send(strAddressBlock)  
  
long Send(BSTR *pbstrAddressBlock)
```

Description

Send a block of addresses to the Move Update server.

Parameters

*strAddressBlock as String*

A string that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

*pbstrAddressBlock*

A pointer to a BSTR that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You may want to experiment with the mrtkINPUT\_BLOCK\_RECORD\_COUNT property. In preliminary tests, we have found the optimal setting to be around 25–50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500–1000. A setting of 25–50 makes the process run very slowly in client-server mode.

Call DoProcess after you have finished sending all of your records to the Move Update Server.

See also

### [DoProcess](#)

See the [CASSTask Properties](#) table for definition of:

- ctRECORD\_COUNT

See the MOVETask Properties table for definition of:

- nINCOALINK\_SHOW\_PROGRESS
- nINCOALINK\_HIDE\_PROGRESS\_AFTER\_PROCESS

See the [MRTK Global Properties](#) table for definition of:

- mrtkINPUT\_BLOCK\_RECORD\_COUNT
- mrtkDELIMITER\_FIELD
- mrtkDELIMITER\_RECORD
- mrtkFIELD\_LIST\_IN

## DoProcess

### Syntax

```
DoProcess  
  
long DoProcess()
```

### Description

Perform NCOA<sup>Link</sup> processing on the records that have been sent to the Move Update server.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Call this function after your final call to Send.

This function will initiate the Move Update process. All records already sent to the Move Update server will be compared against the NCOA<sup>Link</sup> database.

You will not be able send additional records to the server after you call DoProcess.

See also

- [DoProcessEx](#)
- [Send](#)

## DoProcessEx

Syntax

```
DoProcessEx
```

```
long DoProcessEx(long *pbContinue)
```

Description

Perform NCOA<sup>Link</sup> processing on the next 500 records that have been sent to the Move Update server.

Parameters

None

*pbContinue*

Returns value that indicates whether all records have been processed.

Return values

A value that indicates whether all records have been processed (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

Call this function after your final call to Send.

This function will initiate the Move Update process. All records already sent to the Move Update server will be compared against the NCOA<sup>Link</sup> database.

You will not be able send additional records to the server after you call DoProcessEx.

Unlike DoProcess, which processes the all records sent before returning execution to the calling program, this function will process the list in increments of 500 records, so you must use a loop to continuously call it until the returned value is 0.

See also

- [DoProcess](#)
- [Send](#)

## Retrieve

Syntax

```
Retrieve  
  
long Retrieve(BSTR *pbstrAddressBlock)
```

Description

Retrieve processed records from Move Update server.

Parameters

None.

*pbstrAddressBlock*

Returns a block of mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

Return values

A block of mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

The retrieved records will contain fields specified by the mrtkFIELD\_LIST\_OUT property.

See also

See the [MRTK Global Properties](#) table for definition of:

- mrtkINPUT\_BLOCK\_RECORD\_COUNT
- mrtkDELIMITER\_FIELD
- mrtkDELIMITER\_RECORD
- mrtkFIELD\_LIST\_OUT

## Print3553

### Syntax

```
Print3553(bstrPrinterName, bShowPrintSetupDlg)
```

```
long Print3553(BSTR bstrPrinterName, long bShowPrintSetupDlg)
```

### Description

Print the Address Correction report (PS Form 3553).

### Parameters

*bstrPrinterName as String*

The name of the printer to which you wish to print a report.

*bShowPrintSetupDlg as Long*

1 to show the Printer Setup dialog box or 0 to hide it.

*bstrPrinterName*

The name of the printer to which you wish to print a report.

*bShowPrintSetupDlg*

1 to show the Printer Setup dialog box or 0 to hide it.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

If bstrPrinterName is set to an empty string ("") then the default printer will be used to print the form.

If bShowPrintSetupDlg is set to 1 then the value for bstrPrinterName will be ignored.

See also

- [PrintPreview3553](#)
- [PrintReports](#)

## PrintActivationReport

Syntax

```
PrintActivationReport(bstrPrinterName, bShowPrintSetupDlg)  
  
long PrintActivationReport(BSTR bstrPrinterName, long  
bShowPrintSetupDlg)
```

Description

Print the NCOA<sup>Link</sup> account activation form.

Parameters

*bstrPrinterName as String*

The name of the printer to which you wish to print a report.

*bShowPrintSetupDlg as Long*

1 to show the Printer Setup dialog box or 0 to hide it.

*bstrPrinterName*

The name of the printer to which you wish to print a report.

*bShowPrintSetupDlg*

1 to show the Printer Setup dialog box or 0 to hide it.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

If bstrPrinterName is set to an empty string ("") then the default printer will be used to print the form.

If bShowPrintSetupDlg is set to 1, then the value for bstrPrinterName will be ignored.

We recommend that you call ShowMOVEWizard and run the MOVE Wizard to activate your Move Update account. The MOVE Agent Wizard can activate your account instantly via the Internet.

See also

- [PrintPreviewActivationReport](#)
- [ShowMOVEWizard](#)

## PrintNDIForm

Syntax

```
PrintNDIForm(bstrPrinterName, bShowPrintSetupDlg)  
long PrintNDIForm(BSTR bstrPrinterName, long bShowPrintSetupDlg)
```

Description

Print the NCOA<sup>Link</sup> Summary report.

Parameters

*bstrPrinterName as String*

The name of the printer to which you wish to print a report.

*bShowPrintSetupDlg as Long*

1 to show the Printer Setup dialog box or 0 to hide it.

*bstrPrinterName*

The name of the printer to which you wish to print a report.

*bShowPrintSetupDlg*

1 to show the Printer Setup dialog box or 0 to hide it.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

If bstrPrinterName is set to an empty string ("") then the default printer will be used to print the form.

If `bShowPrintSetupDlg` is set to 1, then the value for `bstrPrinterName` will be ignored.

See also

- [PrintPreviewNDIForm](#)
- [PrintReports](#)

## PrintPreview3553

Syntax

```
PrintPreview3553  
long PrintPreview3553()
```

Description

Preview the Address Correction report (PS Form 3553).

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

See also

- [Print3553](#)
- [PrintPreviewReports](#)

## PrintPreviewActivationReport

Syntax

```
PrintPreviewActivationReport  
long PrintPreviewActivationReport()
```

Description

Preview the NCOA<sup>Link</sup> account activation forms.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [\\_COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [\\_COM MRTKTaskLib result code](#).

#### Notes

We recommend that you call ShowMOVEWizard and run the MOVE Wizard to activate your Move Update account. The MOVE Agent Wizard can activate your account instantly via the Internet.

#### See also

- [PrintActivationReport](#)
- [ShowMOVEWizard](#)

## PrintPreviewNDIForm

#### Syntax

```
PrintPreviewNDIForm  
long PrintPreviewNDIForm()
```

#### Description

Preview the NCOA<sup>Link</sup> summary report.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [\\_COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [\\_COM MRTKTaskLib result code](#).

## Notes

## See also

- [PrintNDIForm](#)
- [PrintPreviewReports](#)

## SaveReportAsPDF

### Syntax

```
SaveReportAsPDF(MRTKReportID, bstrFileName, nFileOption)
```

```
long SaveReportAsPDF(long MRTKReportID, BSTR bstrFileName, long  
nFileOption, BSTR *pbstrOutputFileName)
```

### Description

Save a report as a PDF file.

### Parameters

#### *MRTKReportID as Long*

Set to one of the following values:

- 340 – Address Correction summary report (PS Form 3553).
- 353 – NCOA<sup>Link</sup> account activation forms.
- 354 – NCOA<sup>Link</sup> summary report.

#### *bstrFileName as String*

The file name for the saved report.

#### *nFileOption as Long*

Set to one of the following values:

- 0 – Overwrite if the file name specified in bstrFileName already exists.
- 1 – Prompt if the file name specified in bstrFileName already exists.
- 2 – Create new file if the file name specified in bstrFileName already exists. A number will

be appended to `bstrFileName`.

- 3 – Append if the file name specified in `bstrFileName` already exists (not yet implemented).

### *MRTKReportID*

Set to one of the following values:

- 340 – Address Correction summary report (PS Form 3553).
- 353 – NCOA<sup>Link</sup> account activation forms.
- 354 – NCOA<sup>Link</sup> summary report.

### *bstrFileName*

The file name for the saved report.

### *nFileOption*

Set to one of the following values:

- 0 – Overwrite if the file name specified in `bstrFileName` already exists.
- 1 – Prompt if the file name specified in `bstrFileName` already exists.
- 2 – Create new file if the file name specified in `bstrFileName` already exists. A number will be appended to `bstrFileName`.
- 3 – Append if the file name specified in `bstrFileName` already exists (not yet implemented).

### *pbstrOutputFileName*

Returns the name of the saved file.

### Return values

The name of the saved file. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

See also

[SaveReportsAsPDF](#)

## PrintPreviewReports

Syntax

```
PrintPreviewReports  
long PrintPreviewReports()
```

Description

Preview the Address Correction summary (PS Form 3553) and NCOA<sup>Link</sup> summary reports.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

See also

- [PrintPreview3553](#)
- [PrintPreviewNDIForm](#)

## PrintReports

Syntax

```
PrintReports(bstrPrinterName, bShowPrintSetupDlg)  
long PrintReports(BSTR bstrPrinterName, long bShowPrintSetupDlg)
```

Description

Print the Address Correction summary (PS Form 3553) and NCOA<sup>Link</sup> summary reports.

Parameters

*bstrPrinterName as String*

The name of the printer to which you wish to print the reports.

*bShowPrintSetupDlg as Long*

1 to show the Printer Setup dialog box or 0 to hide it.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

If bstrPrinterName is set to an empty string ("") then the default printer will be used to print the form.

If bShowPrintSetupDlg is set to 1, then the value for bstrPrinterName will be ignored.

#### See also

- [Print3553](#)
- [PrintNDIForm](#)

## SaveReportsAsPDF

#### Syntax

```
SaveReportsAsPDF(bstrFileName, nFileOption)  
  
long SaveReportsAsPDF(BSTR bstrFileName, long nFileOption,  
BSTR *pbstrOutputFileName)
```

#### Description

Saves the Address Correction report (PS Form 3553) and the NCOA<sup>Link</sup> summary report as a PDF file.

#### Parameters

*bstrFileName as String*

The file name for the saved report.

*nFileOption as Long*

Set to one of the following values:

- 0 – Overwrite if the file name specified in `bstrFileName` already exists.
- 1 – Prompt if the file name specified in `bstrFileName` already exists.
- 2 – Create new file if the file name specified in `bstrFileName` already exists. A number will be appended to `bstrFileName`.
- 3 – Append if the file name specified in `bstrFileName` already exists (not yet implemented).

### *bstrFileName*

The file name for the saved report.

### *nFileOption*

Set to one of the following values:

- 0 – Overwrite if the file name specified in `bstrFileName` already exists.
- 1 – Prompt if the file name specified in `bstrFileName` already exists.
- 2 – Create new file if the file name specified in `bstrFileName` already exists. A number will be appended to `bstrFileName`.
- 3 – Append if the file name specified in `bstrFileName` already exists (not yet implemented).

### *pbstrOutputFileName*

Returns the name of the saved file.

### Return values

The name of the saved file. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

### See also

[SaveReportAsPDF](#)

## EndTask

### Syntax

```
EndTask  
  
long EndTask()
```

### Description

When finished with the task, clean up and release and resources used.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Call this when you have finished processing and saving all of your address records.

## MOVETask Properties

The MOVETask properties shown below are members of the MRTKTASKLib.MOVETaskPropertyID enumeration. These enum names are used as arguments for the GetProperty and SetProperty functions. If you have added the BCC Architect files, then you can view all of the available enums in the Object Browser.

### nIMRTK\_VERSION

#### Enum Value

212

#### Data Type

String

#### Description

Returns the current BCC Architect server version.

## nINCOA\_BUYER\_NAME

### Enum Value

1743

### Data Type

String

### Description

The name of the person who owns the list being processed with MOVETask.

## nINCOA\_CLIENT\_ID\_LIST

### Enum Value

1879

### Data Type

String

### Description

Returns a list of client IDs and their PAF expiration dates as a single string. The format is [CLIENT\_ID], YYYY-MM-DD. Each ID-date pair is separated from others by a carriage return.

Before you can retrieve this data, you must set nINCOALINK\_BROKER\_ID or nINCOALINK\_ADMIN\_ID and the Broker or Admin password and then call ValidateProperties.

Only Broker and List Administrator accounts can retrieve a list of clients.

The default value is an empty string.

## nINCOA\_CLIENT\_ID\_NAME\_LIST

### Enum Value

7272

### Data Type

String

### Description

Returns a list of client IDs, their names, their PAF expiration dates, and the status of their PAF form as a single string. The format is [CLIENT\_ID], CUSTOMER\_NAME, MM-DD-YYYY, STATUS. Each ID-

name-date-status entry is separated from others by a carriage return.

STATUS can have one of the following values:

- Expired – PAF is expired.
- Pending – PAF signature is pending approval.
- Blank – PAF is up-to-date.

Before you can retrieve this data, you must set `nINCOALINK_BROKER_ID` or `nINCOALINK_ADMIN_ID` and the Broker or Admin password and then call `ValidateProperties`.

Only Broker and List Administrator accounts can retrieve a list of clients.

The default value is an empty string.

#### `nINCOA_CUSTOMER_ADDRESS`

Enum Value

1733

Data Type

String

Description

The address for the client customer processing the list.

#### `nINCOA_CUSTOMER_EMAIL`

Enum Value

1740

Data Type

String

Description

The email address for the client customer processing the list.

#### `nINCOA_CUSTOMER_FAX`

Enum Value

1741

## Data Type

String

## Description

The fax number for the client customer processing the list.

**nINCOA\_CUSTOMER\_LASTLINE**

## Enum Value

1734

## Data Type

String

## Description

The last line of the address for the client customer processing the list.

**nINCOA\_CUSTOMER\_MAILERID**

## Enum Value

1791

## Data Type

String

## Description

The Mailer ID for the customer processing the list.

The default value is an empty string.

**nINCOA\_CUSTOMER\_NAME**

## Enum Value

1736

## Data Type

String

## Description

The name of the client customer processing the list.

## nINCOA\_CUSTOMER\_PHONE

Enum Value

1738

Data Type

String

Description

The phone number for the client customer processing the list.

## nINCOA\_CUSTOMER\_SIC

Enum Value

1747

Data Type

String

Description

The Standard Industrial Classification (now called NAICS) code for the client customer processing the list.

## nINCOA\_CUSTOMER\_TITLE

Enum Value

1737

Data Type

String

Description

The job title for the client customer processing the list.

## nINCOA\_HIGH\_MATCH\_RATE\_REASON

Enum Value

1868

## Data type

String

## Description

Specifies information about the list processed with NCOA<sup>Link</sup>.

The value of this property is stored in the NCOA<sup>Link</sup> job table along with other statistics about the job.

The following table lists the available property values:

Value	Description
A	The input file consists of packages that were matched beyond 18 months in an ANK <sup>Link</sup> file
S	The input file is a Stage 1 file
R	The input file consists of addresses that generated returned mail
""	No reason given (default value)

## nINCOA\_KEEP\_ALTERNATE\_ADDRESS\_LINE

## Enum Value

1880

## Data type

bool

## Description

Specifies whether to retain original address information in addition to the updated address when a moved address is found.

True to return the original address in addition to the new address

Default value is False

## nINCOA\_MAILCLASS

## Enum Value

1742

## Data Type

String

## Description

A single character that represents the class of mailing for the list processed with NCOA:

- A – First Class Mail only
- B – Periodicals only
- C – Standard Mail only
- D – Package Services only
- E – First Class Mail and Periodicals
- F – First Class Mail and Standard Mail
- G – First Class Mail and Package Services
- H – Periodicals and Standard Mail
- I – Periodicals and Package Services
- J – Standard Mail and Package Services
- K – First Class Mail, Periodicals and Standard Mail
- L – First Class Mail, Periodicals and Package Services
- M – First Class Mail, Standard Mail and Package Services
- N – Periodicals, Standard Mail and Package Services
- O – All mailing classes. This is the default value.

## nINCOA\_MOVE\_MONTH\_RANGE

### Enum Value

1746

#### Data Type

Long

#### Description

Determines the maximum number of months ago a move may have occurred in order to change a matched address.

The minimum value is 6

BCC Software is a full service provider, so therefore the maximum value is 48. This is also the default value.

### nINCOA\_MULTI\_NAME\_HANDLE

#### Enum Value

1758

#### Data Type

Long

#### Description

Determines how addresses with multiple names are handled:

- 1 – Search if common last name is found
- 3 – Search using all name found
- 0 – Skip the address

### nINCOALINK\_ADMIN\_ID

#### Enum Value

1702

#### Data Type

String

#### Description

The Move Update User ID for a List Administrator account type.

The default value is an empty string.

## nINCOALINK\_ADMIN\_PASSWORD

### Enum Value

1705

### Data Type

String

### Description

The Move Update password for a List Administrator account type.

The default value is an empty string.

## nINCOALINK\_BROKER\_ID

### Enum Value

1701

### Data Type

String

### Description

The Move Update User ID for a List Broker account type.

The default value is an empty string.

## nINCOALINK\_BROKER\_PASSWORD

### Enum Value

1704

### Data Type

String

### Description

The Move Update password for a List Broker account type.

The default value is an empty string.

## nINCOALINK\_CUSTOMER\_ID

### Enum Value

1700

### Data Type

String

### Description

The Move Update User ID for a Client account type.

The default value is an empty string.

## nINCOALINK\_CUSTOMER\_PASSWORD

### Enum Value

1703

### Data Type

String

### Description

The Move Update password for a Client account type.

The default value is an empty string.

## nINCOALINK\_HIDE\_PROGRESS\_AFTER\_PROCESS

### Enum Value

1761

### Data Type

Long

### Description

Determines whether to hide the Progress screen after the MOVETask finishes processing addresses.

The default value is 0.

## nINCOALINK\_MATCH\_FLAG

### Enum Value

1706

### Data Type

String

Determines what type of moves the service will search for.

### Notes

Use one of the following:

- S = Standard. Business, Individual and Family. This is the default value.
- C= Business and Individual
- B= Business only
- I= Individual only
- R=Individual and Family

## nINCOALINK\_SHOW\_PROGRESS

### Enum Value

1760

### Data Type

Long

### Description

Determines whether to hide the Progress screen while the MOVETask processes addresses.

The default value is 0.

## COM MOVETask Properties Summary Table

MoveTask Properties	Enum Value	Data Type	Default Value	Description
nIMRTK_VERSION	212	String	EMPTY	Returns the current BCC Architect server version.
nINCOA_BUYER_NAME	1743	String	EMPTY	Name of person buying list that was processed with NCOA.
nINCOA_CLIENT_ID_LIST	1879	String	EMPTY	Returns a list of client IDs and their PAF expiration dates as a single string. The format is [CLIENT_ID], YYYY-MM-DD. Each ID-date pair is separated from others by a carriage return. Before you can retrieve this data, you must set nINCOALINK_BROKER_ID or nINCOALINK_CUSTOMER_ID first, then call ValidateProperties.
nINCOA_CLIENT_ID_NAME_LIST	7272	String	EMPTY	Returns a list of client IDs, their names and their PAF expiration dates as a single string. The format is [CLIENT_ID], CUSTOMER_NAME, MMDD-YYYY, STATUS. Each ID-name-date-status entry is separated from others by a carriage return. Before you can retrieve this data, you must set nINCOALINK_BROKER_ID or nINCOALINK_ADMIN_ID and the Broker or Admin password and then call ValidateProperties.
nINCOA_CUSTOMER_ADDRESS	1733	String	EMPTY	The address for the client customer processing the list.
nINCOA_CUSTOMER_EMAIL	1740	String	EMPTY	The email address for the client customer processing the list.

MoveTask Properties	Enum Value	Data Type	Default Value	Description
nINCOA_ CUSTOMER_FAX	1741	String	EMPTY	The fax number for the client customer processing the list.
nINCOA_ CUSTOMER_ LASTLINE	1734	String	EMPTY	The last line of the address for the client customer processing the list.
nINCOA_ CUSTOMER_ MAILERID	1791	String	EMPTY	The Mailer ID for the customer processing the list.
nINCOA_ CUSTOMER_NAME	1736	String	EMPTY	The name of the client customer processing the list.
nINCOA_ CUSTOMER_PHONE	1738	String	EMPTY	The phone number for the client customer processing the list.
nINCOA_ CUSTOMER_SIC	1747	String	EMPTY	The Standard Industrial Classification (now called NAICS) code for the client customer processing the list.
nINCOA_ CUSTOMER_TITLE	1737	String	EMPTY	The job title for the client customer processing the list.

MoveTask Properties	Enum Value	Data Type	Default Value	Description
nINCOA_HIGH_MATCH_RATE_REASON	1868	String	EMPTY	<p>Specifies information about the list processed with NCOA<sup>Link</sup>. The value of this property is stored in the NCOA<sup>Link</sup> job table along with other statistics about the job. The following table lists the available property values:</p> <ul style="list-style-type: none"> <li>• A – The input file consists of packages that were matched beyond 18 months in an ANKlink file.</li> <li>• S – The input file is a Stage 1 file.</li> <li>• R – The input file consists of addresses that generated returned mail.</li> <li>• "" – No reason given.</li> </ul>
nINCOA_KEEP_ALTERNATE_ADDRESS_LINE	1880	Bool	False	<p>If True, retains original address information in addition to the updated address when a moved address is found.</p>

MoveTask Properties	Enum Value	Data Type	Default Value	Description
nINCOA_MAILCLASS	1742	String	"O"	<p>A single character representing the class of mailing for the list processed with NCOA:</p> <ul style="list-style-type: none"> <li>• A - First Class Only</li> <li>• B - Periodicals Only</li> <li>• C - Standard Mail Only</li> <li>• D - Package Services Only</li> <li>• E - First Class And Periodicals</li> <li>• F - First Class And Standard</li> <li>• G - First Class And Package</li> <li>• H - Periodicals And Standard</li> <li>• I - Periodicals And Package</li> <li>• J - Standard And Package</li> <li>• K - First Class, Periodicals, Standard</li> <li>• L - First Class, Periodicals, Package</li> <li>• M - First Class, Standard, Package</li> <li>• N - Periodicals, Standard, Package</li> <li>• O - All</li> </ul>

MoveTask Properties	Enum Value	Data Type	Default Value	Description
nINCOA_MOVE_MONTH_RANGE	1746	Long	18/48	Determines the maximum number of months ago a move may have occurred in order for the address change to be applied. The minimum value is 6. The maximum value is 18 for a limited service provider and 48 for a full service provider. BCC Software is a full service provider.
nINCOA_MULTI_NAME_HANDLE	1758	Long	1	Determines how addresses with multiple names are handled: <ul style="list-style-type: none"> <li>• 1 – Search if common last name is found</li> <li>• 3 – Search using all name found</li> <li>• 0 – Skip the address</li> </ul>
nINCOALINK_ADMIN_ID	1702	String	EMPTY	NCOA User ID for List Administrator account type
nINCOALINK_ADMIN_PASSWORD	1705	String	EMPTY	NCOA password for List Administrator account type
nINCOALINK_BROKER_ID	1701	String	EMPTY	NCOA User ID for Broker account type
nINCOALINK_BROKER_PASSWORD	1704	String	EMPTY	NCOA password for Broker account type
nINCOALINK_CUSTOMER_ID	1700	String	EMPTY	NCOA User ID for Client account type

MoveTask Properties	Enum Value	Data Type	Default Value	Description
nINCOALINK_CUSTOMER_PASSWORD	1703	String	EMPTY	NCOA password for Client account type
nINCOALINK_HIDE_PROGRESS_AFTER_PROCESS	1761	Long	FALSE	TRUE to hide progress dialog after NCOA job is complete
nINCOALINK_MATCH_FLAG	1706	String	"S"	<p>Determines what type of moves the service will search for.</p> <p>Use one of the following:</p> <ul style="list-style-type: none"> <li>• S = Standard. Business, Individual and Family. This is the default value.</li> <li>• C= Business and Individual</li> <li>• B= Business only</li> <li>• I= Individual only</li> <li>• R=Individual and Family</li> </ul>
nINCOALINK_SHOW_PROGRESS	1760	Long	FALSE	TRUE to show the NCOA <sup>Link</sup> progress dialog box.

## The COM PresortTask Object for Presorting Mailings

The PresortTask object performs a sorts a list of addresses according to USPS standards.

The PresortTask offers a flexible interface for postal presorting. You can specify the information returned, the number of records to process, how user interface windows are displayed and several other properties (see the PresortTask Properties table for a complete description).

In addition to presorting functionality, PresortTask can also incorporate address correction. For instance, you can either process a mailing list via the CASSTask object and then use PresortTask, or you can set the PresortTask to perform address correction on a list before presorting it.

The PresortTask should be created via the BCC Architect object factory MRTKObjFactory.

## PresortTask Functions

The PresortTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

### PrepareTask

#### Syntax

```
PrepareTask  
  
long PrepareTask()
```

#### Description

Initialize and prepare the PresortTask object.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You must call this function before any of the other functions or properties of PresortTask. Failing to do so will cause subsequent function calls to fail.

### GetProperty

#### Syntax

```
GetProperty(mrtkPropertyID)  
  
long GetProperty(long mrtkPropertyID, VARIANT *pVal)
```

#### Description

Retrieve a PresortTask property value.

## Parameters

*mrtkPropertyID as Long*

The ID of the property to retrieve.

*mrtkPropertyID*

The ID of the property to retrieve.

*pVal*

Returns the value of the property.

## Return values

The value of the property (as Variant). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

### See also

See the [PresortTask Properties](#) table for a list of property IDs.

## SetProperty

### Syntax

```
SetProperty(mrtkPropertyID, value)
```

```
long SetProperty(long mrtkPropertyID, VARIANT value)
```

### Description

Set a PresortTask property.

### Parameters

*mrtkPropertyID as Long*

The ID of the property to set.

*value as Variant*

The value of the property to set.

*mrtkPropertyID*

The ID of the property to set.

### Value

The value of the property to set.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

#### See also

See the [PresortTask Properties](#) table for a list of property IDs.

## ValidateProperties

### Syntax

```
ValidateProperties  
long ValidateProperties()
```

### Description

Verify that the task is set up correctly and ready to run.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

This function verifies that the necessary input fields have been included for the selected sort. For example, if an Enhanced Carrier Route (LOT) sort is chosen for a Standard Mail class mailing, then the LOT number field would need to be included as one of the input fields.

If you have specified that the mailing list should be address corrected before postal presorting, then the requirements for the CASSTask will also have to be met.

This function needs to be called before calling Send.

## See also

See the [PresortTask Properties](#) table for a list of property IDs.

## Send

### Syntax

```
Send(strAddressBlock)
```

```
long Send(BSTR *pbstrAddressBlock)
```

### Description

Send a block of addresses to PresortTask.

### Parameters

*strAddressBlock as String*

A string that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

*pbstrAddressBlock*

A BSTR that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You may want to experiment with the mrtkINPUT\_BLOCK\_RECORD\_COUNT property. In preliminary tests, we have found the optimal setting to be around 25–50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500-1000. A setting of 25-50 makes the process run very slowly in client-server mode. Call DoSort after you have finished sending all of your records to the PresortTask object.

If your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix), use the SendV function instead.

See also

- [SendV](#)
- [DoSort](#)
- [ShowPresortWizard](#)

## SendV

Syntax

```
SendV(strAddressBlock)  
long SendV(BSTR *pbstrAddressBlock)
```

Description

Send a block of addresses to PresortTask.

Parameters

*strAddressBlock as String*

A string that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

*pbstrAddressBlock*

A BSTR that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You may want to experiment with the mrtkINPUT\_BLOCK\_RECORD\_COUNT property. In preliminary tests, we have found the optimal setting to be around 25-50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500-1000. A setting of 25-50 makes the process run very slowly in client-server mode. Call DoSort after you have finished sending all of your records to the PresortTask object.

You must use this function if your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix). Otherwise, you can use either this function or [Send](#).

See also

- [Send](#)
- [DoSort](#)
- [ShowPresortWizard](#)

## DoSort

Syntax

```
DoSort  
long DoSort()
```

Description

Perform the configured mail sort on the records sent to PresortTask.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

Call this function after your final call to [Send](#).

See also

- [Send](#)
- [ShowPresortWizard](#)

See the [PresortTask Properties](#) table for definition of:

- `ptSHOW_SORT_PROGRESS`

## Retrieve

### Syntax

```
Retrieve(strAddressBlock)  
long Retrieve(BSTR *pbstrAddressBlock)
```

### Description

Retrieve a block of addresses in sorted order from PresortTask.

### Parameters

#### *StrAddressBlock as String*

Returns block of `ptRECORD_COUNT_PER_RECEIVE` addresses that are separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`.

#### *pbstrAddressBlock*

Returns block of `ptRECORD_COUNT_PER_RECEIVE` addresses that are separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Retrieves addresses with fields specified by the `mrtkFIELD_LIST_OUT` property.

Addresses will be corrected prior to sorting if you set the `ptCASS_CERTIFY_FIRST` property to True with the `SetProperty` function.

If your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix), use the `RetrieveV` function instead.

### See also

- [RetrieveV](#)

See the [MRTK Global Properties](#) table for definition of:

- mrtkFIELD\_LIST\_OUT
- mrtkDELIMITER\_RECORD
- mrtkDELIMITER\_FIELD

See the [PresortTask Properties](#) table for definition of:

- ptCASS\_CERTIFY\_FIRST
- ptRECORD\_COUNT\_PER\_RECEIVE

## RetrieveV

### Syntax

```
RetrieveV  
  
long RetrieveV(BSTR *pbstrAddressBlock)
```

### Description

Retrieve a block of addresses in sorted order from PresortTask.

### Parameters

None.

*pbstrAddressBlock*

Returns block of mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

### Return values

A block of mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Retrieves addresses with fields specified by the mrtkFIELD\_LIST\_OUT property.

Addresses will be corrected prior to sorting if you set the ptCASS\_CERTIFY\_FIRST property to True with the SetProperty function.

You must use this function if your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix). Otherwise, you can use either this function or `Retrieve`.

See also

### [Retrieve](#)

See the [MRTK Global Properties](#) table for definition of:

- `mrtkFIELD_LIST_OUT`
- `mrtkDELIMITER_RECORD`
- `mrtkDELIMITER_FIELD`

See the [PresortTask Properties](#) table for definition of:

- `ptCASS_CERTIFY_FIRST`
- `ptRECORD_COUNT_PER_RECEIVE`

## ShowPresortWizard

Syntax

VB:

```
ShowPresortWizard  
long ShowPresortWizard()
```

Description

Displays the PRESORT Agent Wizard, which allows the user to configure the presort settings manually.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

Call this function before calling Send.

The wizard needs to be called only the first time a presort is performed.

Using the wizard, the user can specify mail sort options and save these options as a template. This template can then be accessed with the MRTK global property mrtkTEMPLATE\_NAME\_TO\_USE.

The ptENABLE\_CASS\_PROCESSING property must be set to True if you want to print or preview the Address Correction report.

The page that allows the user to select the reports to print will only be visible if either ptSHOW\_SORT\_PROGRESS is set to False or ptHIDE\_SORT\_PROGRESS\_AFTER\_SORT is set to True before calling this function.

## See also

See the [MRTK Global Properties](#) table for definition of:

- mrtkTEMPLATE\_NAME\_TO\_USE

See the [PresortTask Properties](#) table for definition of:

- ptENABLE\_CASS\_PROCESSING
- ptSHOW\_SORT\_PROGRESS
- ptHIDE\_SORT\_PROGRESS\_AFTER\_SORT

## ShowReportWizard

### Syntax

```
ShowReportWizard  
long ShowReportWizard()
```

### Description

Display the Print Reports dialog box. [OBSOLETE]

### Parameters

None

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This function should be called after ShowPresortWizard.

**NOTE** This function is obsolete. The reports can be printed from PRESORT Agent Wizard.

### ShowLabelWizard

#### Syntax

```
ShowLabelWizard  
  
long ShowLabelWizard()
```

#### Description

Launches the Label Wizard, where the user can design and print envelopes and labels.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This function should be called after performing a presort.

This function is not currently implemented.

### GetPropertySummary

#### Syntax

```
GetPropertySummary(mrtkPropertyID)  
  
long GetPropertySummary(long mrtkPropertyID, BSTR *pVal)
```

#### Description

Returns the display string for a property.

## Parameters

*mrtkPropertyID as Long*

The ID of the property for which you want summary information.

*mrtkPropertyID*

The ID of the property for which you want summary information.

*pVal*

Returns a BSTR that contains a description of the property specified by mrtkPropertyID.

## Return values

A string that contains a description of the property specified by mrtkPropertyID (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function can be used to display the description of a property on screen to the user.

## See also

See the [PresortTask Properties](#) table for a list of property IDs

## PrintReport

### Syntax

```
PrintReport(MRTKReportID, bstrPrinterName, bShowPrintSetupDlg)
```

```
long PrintReport(long MRTKReportID, BSTR bstrPrinterName, long  
bShowPrintSetupDlg)
```

### Description

Prints a postal presort report.

### Parameters

*MRTKReportID as Long*

The ID of the report you wish to print.

*bstrPrinterName as String*

The name of the printer on which you wish to print a report.

### *bShowPrintSetupDlg as Long*

1 to show the Printer Setup dialog box or 0 to hide it.

### *MRTKReportID*

The ID of the report you wish to print.

### *bstrPrinterName*

The name of the printer on which you wish to print a report.

### *bShowPrintSetupDlg* –

1 to show the Printer Setup dialog box or 0 to hide it.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

If *bstrPrinterName* is set to an empty string, ("") then the default printer will be used to print the indicated report.

If *bShowPrintSetupDlg* is set to 1 then *bstrPrinterName* will be ignored.

Calling this function will override the selections made within the Presort Wizard.

To print the reports as specified in the Presort Wizard, call `PrintPresortReports`.

### See also

- [PrintPresortReports](#)
- See the [MRTKReportID](#) table for a list of report IDs

## PreviewReport

### Syntax

```
PreviewReport (MRTKReportID)
```

```
long PreviewReport (long MRTKReportID)
```

### Description

Displays a postal presort report on the user's screen.

## Parameters

### *MRTKReportID as Long*

The ID of the report you wish to preview.

### *MRTKReportID*

The ID of the report you wish to preview.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

Each call to this function will create a new window for the specified report. Hence, multiple calls to PreviewReport will result in multiple preview windows being displayed.

Calling this function will override the selections made within the Presort Wizard.

To preview the reports specified in the Presort Wizard, call PreviewPresortReports.

## See also

- [PreviewPresortReports](#)
- See the [MRTKReportID](#) table for a list of report Ids

## SaveReportAsPDF

### Syntax

```
SaveReportAsPDF(MRTKReportID, bstrFileName, nFileOption)  
  
long SaveReportAsPDF(long MRTKReportID, BSTR bstrFileName,  
long nFileOption, BSTR *pbstrOutputFileName)
```

### Description

Save a postal presort report as a PDF file.

### Parameters

#### *MRTKReportID as Long*

The ID of the report you wish to preview.

*bstrFileName as String*

The name of the file to save.

*nFileOption as Long*

Determines what to do when the file name already exists:

- 0 – Overwrite if file with file name specified in *bstrFileName* already exists.
- 1 – Prompt if file with file name specified in *bstrFileName* already exists.
- 2 – Create new file if file with name specified in *bstrFileName* already exists; a number will be appended to *bstrFileName*.
- 3 – Append if file with name specified in *bstrFileName* already exists (not yet implemented).
- MRTKReportID – The ID of the report you wish to preview.

*bstrFileName*

The name of the file to save.

- *nFileOption* – Determines what to do when the file name already exists:
- 0 – Overwrite if file with file name specified in *bstrFileName* already exists.
- 1 – Prompt if file with file name specified in *bstrFileName* already exists.
- 2 – Create new file if file with name specified in *bstrFileName* already exists; a number will be appended to *bstrFileName*.
- 3 – Append if file with name specified in *bstrFileName* already exists (not yet implemented).

*pbstrOutputFileName*

Returns the name of the saved file.

## Return values

The name of the saved file. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Calling this function will override the selections made within the Presort Wizard.

To save the reports as specified in the Presort Wizard, call `SaveReportsAsPDF`.

#### See also

- [SaveReportsAsPDF](#)
- See the [MRTKReportID](#) table for a list of report IDs

## PrintPresortReports

#### Syntax

VB:

```
PrintPresortReports
```

```
long PrintPresortReports()
```

#### Description

Prints the reports specified in the PresortTask properties

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This function prints the reports selected within the Presort Wizard or whose properties have been set to True with `SetProperty`.

Unless you change the relevant properties with `PrintReport`, a call to this function will print all reports that were selected the last time the Presort Wizard was run or whose properties were last set to True.

Setting `ptPRINT_ALL_REPORTS` equal to True will set the properties of each report to True. This function will then print all reports.

If the reports have been set to be previewed or saved as a PDF file in the Presort Wizard, then they will be previewed or saved as a PDF instead of printed when calling this function.

See also

### [PrintReport](#)

See the [PresortTask Properties](#) table for definition of:

- ptPRINT\_ALL\_REPORTS
- ptPRINT\_POSTAGE
- ptPRINT\_QUALIFICATION
- ptPRINT\_PRESORT
- ptPRINT\_TRAYLABELS
- ptPRINT\_CASS
- ptPRINT\_ZONE

### PreviewPresortReports

Syntax

```
PreviewPresortReports  
long PreviewPresortReports()
```

Description

Preview the reports specified in the PresortTask properties.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function previews the reports selected within the Presort Wizard or whose properties have been set to True with SetProperty.

Unless you change the relevant properties with PreviewReport, a call to this function will print all reports that were selected the last time the Presort Wizard was run or whose properties were last set to True.

Setting ptPRINT\_ALL\_REPORTS equal to True will set the properties of each report to True and hence a call to this function will preview all reports.

The previewed reports will display concatenated in a single window.

See also

### [PreviewReport](#)

See the [PresortTask Properties](#) table for definition of:

- ptPRINT\_ALL\_REPORTS
- ptPRINT\_POSTAGE
- ptPRINT\_QUALIFICATION
- ptPRINT\_PRESORT
- ptPRINT\_TRAYLABELS
- ptPRINT\_CASS
- ptPRINT\_ZONE

## SaveReportsAsPDF

Syntax

```
SaveReportsAsPDF(bstrFileName, nFileOption)  
  
long SaveReportsAsPDF(BSTR bstrFileName, long nFileOption,  
BSTR *pbstrOutputFileName)
```

Description

Save the reports specified in the PresortTask properties as a single PDF file.

## Parameters

### *bstrFileName as String*

The name of the file to save.

### *nFileOption as Long*

Determines what to do when the file name already exists:

- 0 – Overwrite if file with file name specified in *bstrFileName* already exists.
- 1 – Prompt if file with file name specified in *bstrFileName* already exists.
- 2 – Create new file if file with name specified in *bstrFileName* already exists; a number will be appended to *bstrFileName*.
- 3 – Append if file with name specified in *bstrFileName* already exists (not yet implemented).

### *bstrFileName*

The name of the file to save.

### *nFileOption*

Determines what to do when the file name already exists:

- 0 – Overwrite if file with file name specified in *bstrFileName* already exists
- 1 – Prompt if file with file name specified in *bstrFileName* already exists
- 2 – Create new file if file with name specified in *bstrFileName* already exists; a number will be appended to *bstrFileName*
- 3 – Append if file with name specified in *bstrFileName* already exists (not yet implemented)

### *pbstrOutputFileName*

Returns the name of the saved file.

## Return values

The name of the saved file. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This function saves the reports selected within the Presort Wizard or whose properties have been set to True with SetProperty.

Unless you change the relevant properties, a call to this function will save all reports that were selected the last time the Presort Wizard was run or whose properties were last set to True.

Setting ptPRINT\_ALL\_REPORTS equal to True will set the properties of each report to True and hence a call to this function will save all reports.

The reports will all be saved as a single file.

#### See also

#### [SaveReportAsPDF](#)

See the [PresortTask Properties](#) table for definition of:

- ptPRINT\_ALL\_REPORTS
- ptPRINT\_POSTAGE
- ptPRINT\_QUALIFICATION
- ptPRINT\_PRESORT
- ptPRINT\_TRAYLABELS
- ptPRINT\_CASS
- ptPRINT\_ZONE

#### AbortTask

##### Syntax

```
AbortTask  
  
long AbortTask()
```

##### Description

Exit the task early.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### EndTask

##### Syntax

```
EndTask
```

```
long EndTask()
```

##### Description

When finished with the task, clean up and release and resources used.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

Call this when you have finished sorting and retrieving all of your address records.

## PresortTask Properties

The PresortTask properties shown below are members of the MRTKTASKLib.PresortTaskPropertyID enumeration. These enum names are used as arguments for the GetProperty and SetProperty functions. If you have added the BCC Architect files, then you can view all of the available enums in the Object Browser.

### ptACS\_METHOD

#### Enum Value

7061

## Data Type

Long

## Description

Determines which service type code is used in the IM Barcode. See exhibit.

- 0 – None. This is the default value.
- 1 – ManualCorrection
- 2 – ASR
- 3 – CSR
- 4 – PeriodicalASRwithASE
- 5 – PeriodicalACS
- 6 – TraditionalASROption1
- 7 – TraditionalCSROption1
- 8 – TraditionalASROption2
- 9 – TraditionalCSROption2
- 10 – TraditionalPeriodicalACS
- 11 – ASROption2
- 12 – RSR
- 13 – TRSR
- 14 – TraditionalRSR
- 15 - TraditionalTRSR
- 16 – CSROption2
- 17 – CSR1SD

- 18 – CSR2SD
- 19 – TraditionalCSROption1SD
- 20 – TraditionalCSROption2SD
- 21 – ASR1BM
- 22 – ASR2BM
- 23 – CSR1BM

For an overview of Service Type Identifiers (STIDs), see the [Service Type Identifier \(STID\) Table](#) on the [USPS PostalPro](#) website.

#### ptALWAYS\_USE\_PERIODICALS\_FSS\_PREP

Enum Value

7166

Data Type

String

Description

When preparing a Periodicals mailing, will always use FSS if available.

The default value for this property is 0.

- 0 = do not automatically use FSS for periodicals
- 1 = automatically use FSS for periodicals

#### ptANNUAL\_NONSUBSCRIBER\_THRESHOLD\_EXCEEDED

Enum Value

7111

Data Type

bool

## Description

Determines whether the in-county rates are applied to periodicals.

Applies only to periodicals.

Once a periodical has exceeded the annual 10% threshold of mailing to non-subscribers, they can no longer claim in-county rates for those addresses.

Default value is "false"

Set to "true" to indicate the annual threshold has been exceeded.

## ptAPPLY\_BREAK\_MARK\_INDICATOR

### Enum Value

7096

### Data Type

String

## Description

Specify where to apply a break mark indicator for bundles, trays, or pallets.

- 0 = apply break mark on first piece
- 1 = apply break mark on last piece
- The default value for this property is 0.

## ptAPPLY\_PARCEL\_SURCHARGE

### Enum Value

7064

### Data Type

BOOL

## Description

This property will trigger application of the parcel surcharge for the current mailing, if package is not barcoded or if package weighs less than two ounces.

The property has no effect on classes other than 1<sup>st</sup> class. For other classes, it is ignored.

The default value is FALSE.

### ptBREAK\_MARK\_IND\_BUNDLE

#### Enum Value

7093

#### Data Type

String

#### Description

Specifies break mark indicator to use for bundle breaks.

The default value for this property is "#".

### ptBREAK\_MARK\_IND\_CONTAINER

#### Enum Value

7094

#### Data Type

String

#### Description

Specifies the break mark indicator to use for container breaks.

The default value for this property is '##'.

### ptBREAK\_MARK\_IND\_PALLET

#### Enum Value

7095

#### Data Type

String

#### Description

Specifies the break mark to use for pallet breaks.

The default value for this property is '###'.

## ptCASS\_CERTIFY\_FIRST

### Enum Value

312

### Data Type

BOOL

### Description

Determines whether to perform address correction prior to presorting.

The default value is FALSE.

## ptCASS\_PROCESS\_DATE\_AUTOMATION

### Enum Value

6761

### Data Type

String

### Description

Specifies the CASS processing date to print on the postage statement for automation rate pieces.

The default value is an empty string.

## ptCASS\_PROCESS\_DATE\_ECR

### Enum Value

6762

### Data Type

String

### Description

Specifies the CASS processing date to print on the postage statement for ECR rate pieces.

The default value is an empty string.

## ptCOMBINE\_RESIDUAL\_PIECES

### Enum Value

7151

### Data Type

Boolean

### Description

Use this property to combine 1oz and 2oz residual pieces for a blended rate.

The default value for this property is True.

## ptCONFIRM\_TRACKING

### Enum Value

7044

### Data Type

Long

### Description

Determines whether to enable tracking.

Default tracking mechanism is via IMb Tracing.

For Track N Trace service, you must also set PT\_USE\_EASYTRACK to 1.

The default value is 0, not tracking.

Available property values:

Value	Description
0	No tracking
1	Tracking enabled

Limited to First Class Machinable Letters, Flats and Postcards (Automation). Or, Standard Mail Machinable Letters or Flats (Automation or Carrier Route).

**NOTE** Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings. It replaces the EasyTrack service. This property

now provides access to Track N Trace.

## ptCONTAINER\_LABEL\_LAYOUT

### Enum Value

6819

### Data Type

Long

### Description

Determines whether sack and tray labels will be printed as Legacy or Intelligent Mail labels.

Intelligent Mail labels are required to qualify for Full Service discounts and services.

Valid values are:

- 0 – Legacy
- 1 – Intelligent Mail

## ptCUSTOM\_BARCODE\_MAILER\_ID

### Enum Value

7134

### Data Type

String

### Description

### Data type

string

### Description

Third party mailer ID to use for IMb Tracing™.

Must be a Mailer ID registered with USPS® for IMb Tracing.

## ptCREATE\_COURTESY\_PALLETS

### Enum Value

7275

### Data Type

BOOL

### Description

This property enables the option to be able to create courtesy pallets, which are a special type of user pallet that can be used to place residual bundles in sacks on a pallet for submission of a mailing under the USPS Seamless Acceptance program. The property applies to the Standard Mail Flats and Bound Printed Matter Flats mail piece types.

Pallet placards for these pallets are marked as Mixed-NDC.

This property is similar to the pwCREATE\_LOW\_VOLUME\_PALLETS property, which is useful when you want to place unpalletized containers on a Mixed-NDC pallet when this pallet is allowed according to USPS regulations for the selected Mail Class and Piece Type.

## ptCUSTOM\_BARCODE\_MAILER\_ID

### Enum Value

7134

### Data Type

String

### Description

Third party mailer ID to use for IMb Tracing™.

Must be a Mailer ID registered with USPS® for IMb Tracing.

## ptDATA\_SERVICES\_CLIENT\_ID\_LIST

### Enum Value

7263

### Data Type

String

#### Description

Returns a list of client IDs for a specific provider ID that uses BCC Architect Track N Trace.

The default value is an empty string.

#### ptDATA\_SERVICES\_CLIENT\_USER

##### Enum Value

7059

##### Data Type

String

#### Description

The name of the client (if any) associated with the current sort template.

#### ptDATA\_SERVICES\_JOB\_ID

##### Enum Value

7062

##### Data Type

String

#### Description

Specifies the job ID associated with the BCC Architect Track N Trace job.

#### ptDATA\_SERVICES\_PASSWORD

##### Enum Value

7002

##### Data Type

String

#### Description

The BCC Architect Data Services password for the Provider account used with BCC Architect Track N Trace.

## ptDATA\_SERVICES\_USER

### Enum Value

7001

### Data Type

String

### Description

The user name for the Data Services Provider account with BCC Architect Track N Trace.

## ptDrop\_SHIP\_ADC\_ZIPS

### Enum Value

7238

### Data Type

String

### Description

Contains a list of ADC facilities for Plant-Verified Drop Shipment, separated by commas.

The format for each facility is the 3-Digit ZIP Code, optionally followed by a colon and minimum weight in pounds.

If a minimum weight is not supplied, the default minimum weight will be used.

The default value is an empty string.

## ptDROP\_SHIP\_BMC\_ZIPS

### Enum Value

626

### Data Type

String

### Description

This property is obsolete. Use ptDROP\_SHIP\_NDC\_ZIPS instead.

## ptDROP\_SHIP\_NDC\_ZIPS

### Enum Value

626

### Data Type

String

### Description

Contains a list of NDC facilities for Plant-Verified Drop Shipment, separated by commas.

The format for each facility is the 3- or 5-Digit ZIP Code, optionally followed by a colon and the minimum weight in pounds.

If a minimum weight is not supplied, the default minimum weight will be used.

The default value is an empty string.

## ptDROP\_SHIP\_SCF\_ZIPS

### Enum Value

618

### Data Type

String

### Description

List of SCF facilities for Plant-Verified Drop Shipment, separated by commas.

The format for each facility is the 3-Digit ZIP Code, optionally followed by a colon and minimum weight in pounds.

If a minimum weight is not supplied, the default minimum weight will be used.

The default value is an empty string.

## ptENABLE\_CASS\_PROCESSING

### Enum Value

307

### Data Type

BOOL

#### Description

Determines whether to enable printing PS Form 3553 in Presort Wizard.

The default value is TRUE.

#### ptENABLE\_TEMPLATES

#### Enum Value

308

#### Data Type

BOOL

#### Description

Determines whether to display the list of saved Presort templates in the Presort Wizard.

The default value is TRUE.

#### ptEXCEPTIONAL\_DISPATCH\_ZIPS

#### Enum Value

6739

#### Data Type

String

#### Description

Contains a list of 5-digit ZIP Codes delimited by commas that will be used for Periodicals Exceptional dispatch.

The default value is an empty string.

#### ptFIRM\_BUNDLE\_MIN

#### Enum Value

6721

#### Data Type

Long

### Description

Determines the minimum number of pieces required to make a firm bundle.

Enable firm bundles with the `ptUSE_FIRM_BUNDLES` property.

Firm bundles require an add-on to BCC Architect to function.

Firm bundles only apply to Periodicals mailings.

The default value is 2.

### `ptFORCE_WALK_SEQUENCE_SATURATION`

#### Enum Value

815

#### Data Type

BOOL

### Description

Determines whether to apply saturation rates to all ECR pieces without verifying density requirements.

The default value is FALSE.

### `ptFORM_NAME`

#### Enum Value

300

#### Data Type

String

### Description

The name of the list that you are processing.

The default value is "Mailing List".

### `ptHIDE_SORT_PROGRESS_AFTER_SORT`

#### Enum Value

304

## Data Type

BOOL

## Description

Determines whether to hide the progress dialog box after presorting is complete.

The default value is FALSE.

## ptIM\_BARCODE\_MAILER\_ID\_CODE

## Enum Value

6831

## Data Type

Long

## Description

Determines which Mailer ID to use:

- 0 – List owner as specified by ptMAIL\_OWNER\_MAILER\_ID. This is the default value.
- 1 – Mailing agent as specified by ptMAILING\_AGENT\_MAILER\_ID.

## ptIM\_CONTAINER\_SEQUENCE\_LAST

## Enum Value

6826

## Data Type

Long

## Description

Contains the last sequence number used for Intelligent Mail barcodes on a tray or sack.

You cannot set this property.

If no sequence numbers have been used previously, this property will return 0.

The default value is 0.

## ptIM\_CONTAINER\_SEQUENCE\_START

### Enum Value

6823

### Data Type

Long

### Description

The first sequence number for Intelligent Mail barcodes on trays or sacks when you specify manual sequencing in ptIM\_SEQUENCING\_METHOD.

The default value is 1.

## ptIM\_EINDUCTION

### Enum Value

7234

### Data Type

bool

### Description

Indicates whether the Intelligent Mail mailing uses eInduction.

The default value for this property is False.

This property can be set only if you have the Palletization and Mail.dat add-ons and have registered the keys. The keys must be current.

Drop shipment is required. If that condition is met, this property can be set for any pallet-level container that qualifies for Full Service Intelligent Mail. Container barcodes (IMcb) are required and must be unique within a 45-day period.

BCC Architect supports eInduction for the following mail classes:

Sort Class	Piece Type
Periodicals	Letters, Flats
Standard Mail	Letters, Flats
Bound Printed Matter (Package Services)	Flats (barcoded)

## ptIM\_EINDUCTION\_ACCEPT\_MISSHIPPED

### Enum Value

7235

### Data Type

bool

### Description

Indicates whether there is support for misshipped containers in the eInduction mailing.

This property is optional.

## ptIM\_EINDUCTION\_FAST\_SCHEDULER\_ID

### Enum Value

7236

### Data Type

String

### Description

Specifies the Fast Scheduler ID.

The default value for this property is an empty string. If you want to use FAST appointments, provide your FAST\_SCHEDULER\_ID, and log-in to the FAST site to finalize your appointment details .

## ptIM\_PALLET\_SEQUENCE\_LAST

### Enum Value

6827

### Data Type

Long

### Description

Contains the last sequence number used for Intelligent Mail barcodes on a pallet.

You cannot set this property.

If no sequence numbers have been used previously, this property will return 0.

The default value is 0.

### ptIM\_PALLET\_SEQUENCE\_START

Enum Value

6824

Data Type

Long

Description

The first sequence number for Intelligent Mail barcodes on pallets when you specify manual sequencing in ptIM\_SEQUENCING\_METHOD.

The default value is 1.

### ptIM\_PIECE\_SEQUENCE\_LAST

Enum Value

6825

Data Type

Long

Description

Contains the last sequence number used for Intelligent Mail barcodes on a mail piece.

You cannot set this property.

If no sequence numbers have been used previously, this property will return 0.

The default value is 0.

### ptIM\_PIECE\_SEQUENCE\_START

Enum Value

6822

Data Type

Long

### Description

The first sequence number for Intelligent Mail barcodes on mail pieces when you specify manual sequencing in `ptIM_SEQUENCING_METHOD`.

The default value is 1.

### `ptIM_SEQUENCING_METHOD`

#### Enum Value

6821

#### Data Type

Long

### Description

Determines the sequencing method for the Intelligent Mail barcode on mail pieces, containers and pallets.

Valid values are:

- 0 – A single number will be used for all barcodes. You will not qualify for Full Service discounts and services with this value.
- 1 – Automatic numbering based on previously used values and valid ranges. This is the default value.
- 2 – Manual numbering, beginning with the value specified in `ptIM_PIECE_SEQUENCE_START`.
- 3 – Sequence number will be based on the value in the field, `FLD_IM_PIECE_SEQUENCE_ID`.

### `ptINCOUNTY_ZIPS`

#### Enum Value

6746

#### Data Type

String

#### Description

List of 5-digit ZIP Codes separated by commas that will be treated as in-county.

The default value is an empty string.

#### ptJOB\_ID

##### Enum Value

6832

##### Data Type

String

#### Description

Specifies the Job ID, which is printed on the Qualification, Container Listing, and Mail Sort Summary reports.

This property should be used in preference to the `rwPRINT_PRESORT_MAILER_ID` property.

#### ptLABELS\_BARCODE

##### Enum Value

342

##### Data Type

BOOL

#### Description

Determines if a DP Barcode is required for printing labels after a presort.

This property is read only and can only be retrieved after calling `DoSort`.

The default value is `FALSE`.

#### ptLABELS\_ENDORSEMENT

##### Enum Value

343

##### Data Type

BOOL

#### Description

Determines if an endorsement line is required for printing labels after a presort.

This property is read only and can only be retrieved after calling DoSort.

The default value is FALSE.

#### ptLEGACY\_DDU\_SUPPORT

#### Enum Value

7056

#### Data Type

BOOL

#### Description

This property if TRUE, will provide DDU discount for qualified pieces. PS-8125 reports and PS-3602-C reports will not be generated.

If this property is FALSE, drop shipment paperwork will only be created for a particular DDU if the mailing has the minimum number of pieces necessary to create a container or pallet for that DDU's 5-digit ZIP Code or Carrier Routes. Carrier Route data in the Carrier Route field is required for DDU sorting. Density and sequencing requirements still apply for Carrier Route rate eligibility.

Default is FALSE.

#### ptMAIL\_CONTENT

#### Enum Value

7177

#### Data Type

Integer

#### Description

Specifies whether the mailpiece content is eligible for Election Mail or Political Mail discounts.

Possible values:

- 0 – No discount is applied; the mailpiece content does not qualify as Election Mail or Political Mail.

- 1 – Official Election Mail
- 2 – Political Campaign Mail

Default value is 0.

#### ptMAIL\_OWNER\_CRID

##### Enum Value

6841

##### Data Type

String

##### Description

The Customer Registration ID for the owner of this mailing.

The default value is an empty string.

#### ptMAIL\_OWNER\_MAILER\_ID

##### Enum Value

6828

##### Data Type

String

##### Description

Contains the Mailer ID for the owner of the mailing, regardless of whether they are sorting and sending it.

You can specify multiple Mailer IDs, separated by commas.

Additional Mailer IDs will be used if you selected Automatic as the ptIM\_SEQUENCING\_METHOD and the first Mailer ID does not have enough unused sequence numbers to cover all mail pieces in the mailing. All other sequencing methods will only use the first Mailer ID.

The default value is an empty string.

## ptMAILDAT\_CONTACT\_EMAIL

Enum Value

6757

Data Type

String

Description

A string of at most 60 characters that specifies the Mail.dat contact email address.

The default value is an empty string.

## ptMAILDAT\_CONTACT\_NAME

Enum Value

6755

Data Type

String

Description

An alphanumeric string of at most 30 characters that specifies the Mail.dat contact name.

The default value is an empty string.

## ptMAILDAT\_CONTACT\_PHONE

Enum Value

6756

Data Type

String

Description

A ten-digit number that specifies the Mail.dat contact phone.

## ptMAILDAT\_CREATE\_PBC

Enum Value

7127

#### Data Type

Bool

#### Description

PBC (Piece Barcode) files are now created along with other mail.dat files if this property is TRUE. Either PBC or PDR must be selected to prevent an error.

The default is FALSE.

#### ptMAILDAT\_CREATE\_PDR

#### Enum Value

7053

#### Data Type

Bool

#### Description

PDR (Piece Detail Records) files are now created along with other mail.dat files if this property is TRUE. Either PBC or PDR must be selected to prevent an error.

The default is FALSE.

#### ptMAILDAT\_INFORMED\_CODE

#### Enum Value

7247

#### Data Type

String

#### Description

Unique code for the Informed Delivery campaign. The USPS uses this to identify the campaign. This code can be associated with more than one mailing, but in that case, the Campaign Title for all mailings must match.

Maximum length: 40

#### ptMAILDAT\_INFORMED\_END

#### Enum Value

7253

**Data Type**

String

**Description**

Date the Informed Delivery campaign is to end. the date:

- Cannot be in the past
- Cannot come before the Start Date
- Must end at least one day after the Start Date

Leave blank when there is no end date. Do not replace with zeroes.

**ptMAILDAT\_INFORMED\_ID****Enum Value**

7256

**Data Type**

String

**Description**

Unique ID individual Informed Delivery campaign. Can be the same ID as the value in the Campaign Code property.

**ptMAILDAT\_INFORMED\_NAME****Enum Value**

7248

**Data Type**

String

**Description**

Descriptive name for your mailing campaign. The name is used in the Informed Delivery email message to identify who sent the mailing.

## ptMAILDAT\_INFORMED\_REPRESENT

### Enum Value

7251

### Data Type

String

### Description

Required for Flats; Optional for Letters.

URL pointer to an image to replace the default scanned grayscale image of the mail piece.

As a best practice, the USPS recommends that the image match the style, look and feel of the actual mail piece, thereby providing a consistent customer experience.

Ideally, the artwork for this image and the Representative image are closely aligned.

File type: .jpg, .jpeg

Maximum dimensions: 500 pixels height; 780 pixels width

Maximum size: 200 KB

Maximum URL length: 250

## ptMAILDAT\_INFORMED\_RIDEALONG

### Enum Value

7249

### Data Type

String

### Description

Required. URL pointer to an image that includes a clear call to action or next steps. To attract the attention of your customers, the USPS recommends that this image be in color. It also recommends that the image include your company logo.

Ideally, the artwork for this image and the Representative image are closely aligned.

File type: .jpg, .jpeg

Maximum dimensions: 200 pixels height; 300 pixels width

Maximum size: 200 KB

Maximum URL length: 250

### ptMAILDAT\_INFORMED\_START

#### Enum Value

7252

#### Data Type

String

#### Description

Date the Informed Delivery campaign is to begin.

The date cannot be in the past. The USPS recommends that a campaign start 3 days before and end 3 days after the target in-home range for the mailing.

Leave blank when there is no start date. Do not replace with zeroes.

### ptMAILDAT\_INFORMED\_TARGETURL

#### Enum Value

7250

#### Data Type

String

#### Description

Required. URL pointer to a website or portal to provide a customer with more information. You can provide a unique URL for each campaign to track a campaign's results.

Maximum URL length: 250

### ptMAILDAT\_INFORMED\_TITLE

#### Enum Value

7246

#### Data Type

String

#### Description

Descriptive name for your mailing campaign. This is for informational purposes and is visible only to the mailer in the Informed Delivery mailer portal.

#### ptMAILDAT\_MAILING\_FACILITY\_ID

##### Enum Value

6753

##### Data Type

String

#### Description

An alphanumeric of at most 10 characters that specifies the Mail.dat mailing facility ID.

The default value is an empty string.

#### ptMAILDAT\_MAILING\_TITLE

##### Enum Value

6854

##### Data Type

String

#### Description

A string that contains the title of the mailing for Mail.dat files.

The default value is an empty string.

#### ptMAILDAT\_PERMIT\_HOLDER\_ID

##### Enum Value

6754

##### Data Type

String

#### Description

A string of at most eight characters that specifies the Mail.dat permit holder ID.

The default value is an empty string.

#### ptMAILDAT\_USER\_LICENSE\_CODE

Enum Value

6752

Data Type

String

Description

A four-character alphanumeric string that starts with a letter and specifies the Mail.dat user license code.

The default value is an empty string.

#### ptMAILDAT\_VERIFICATION\_FACILITY\_NAME

Enum Value

6758

Data Type

String

Description

An alphanumeric string of at most 30 characters that specifies the Mail.dat facility name.

The default value is an empty string.

#### ptMAILDAT\_VERIFICATION\_FACILITY\_ZIP4

Enum Value

6759

Data Type

String

Description

A nine-digit number that specifies the Mail.dat verification facility ZIP+4 Code.

The default value is an empty string.

## ptMAILDAT\_VERSION

### Enum Value

6843

### Data Type

String

### Description

Sets the version of the Mail.dat files that will be generated.

For mailings on or after January 24, 2018, set this to either "17-1" or "18-1".

The default value is "18-1".

**IMPORTANT** Mail.dat versions can change frequently, check with *PostalOne!* and BCC Software Technical Support for more details.

## ptMAILER\_ID\_USED

### Enum Value

6830

### Data Type

String

### Description

Returns the Mailer ID that was used when multiple Mailer IDs are specified.

You cannot set this property.

## ptMAILING\_AGENT\_CRID

### Enum Value

6842

### Data Type

String

### Description

The Customer Registration ID for the company performing the mailing, if different from the mail owner.

The default value is an empty string.

### ptMAILING\_AGENT\_MAILER\_ID

#### Enum Value

6829

#### Data Type

String

### Description

Contains the Mailer ID for the mailing agent if this mailing is being sent on behalf of another organization.

You can specify multiple Mailer IDs, separated by commas. These additional Mailer IDs will be used if you selected Automatic as the ptIM\_SEQUENCING\_METHOD and the first Mailer ID does not have enough unused sequence numbers to cover all mail pieces in the mailing. All other sequencing methods will only use the first Mailer ID.

The default value is an empty string.

### ptMAILING\_IS\_PLUS\_ONE\_MARRIAGE\_MAIL

#### Enum Value

7279

#### Data Type

Bool

### Description

Indicates if the mailing is PlusOne Marriage Mail.

The default value is False.

### ptMANIFEST\_SEQUENCE\_NUMBER

#### Enum Value

7020

#### Data Type

String

#### Description

Used to set the Manifest sequence number on the Manifest report.

The default value is an empty string.

#### ptMOVE\_UPDATE\_DATE

#### Enum Value

6925

#### Data Type

String

#### Description

Contains the date that this entire mailing list was last processed. This will be printed on the postal forms.

You can organize the date in any standard format, i.e. "23 January 2010", "01-23-2010", "Jan. 26, 2010", etc.

The default is an empty string.

#### ptMOVE\_UPDATE\_METHOD

#### Enum Value

6806

#### Data Type

Long

#### Description

Determines which box is checked in the Move Update Method section of the Postage Statement. Also, determines the Move Update method saved via Mail.DAT.

- 0 – None. This is the default value.
- 1 – NCOA

- 2 – ACS
- 3 – Ancillary Service Endorsement
- 4 – Alternative
- 6 – Multiple
- 7 – OneCode ACS
- 8 – n/a Alternative Address Format
- 9 – Automatic (If ptACS\_METHOD is used will select OneCode ACS.)

**IMPORTANT** USPS has retired FastForward. The previously available value "5 – FastForward" is no longer valid as a selection.

#### ptMRTK\_PREFER\_CONDENSED\_REPORTS

Enum Value

6744

Data Type

BOOL

Description

Determines whether to print EZ version of PS 3600 and PS 3602 when possible.

The default value is FALSE.

#### ptMRTK\_VERSION

Enum Value

212

Data Type

String

Description

Returns the current BCC Architect server version.

## ptOWNER\_GHOST\_NUMBER

### Enum Value

534

### Data Type

String

### Description

If the mailing agent uses a permit other than the mail owners', the mail owner or organization must be identified with a number provided by the same post office that issued the permit used for the mailing. This number is used by USPS to track the mail owner when the mail owner does not have a normal mailing permit.

The default value is an empty string.

Not used if the user's role in the mailing is mail owner or if the mail owner's permit is being used.

## ptPALLET\_PLACARD\_LAYOUT

### Enum Value

6820

### Data Type

Long

### Description

Determines the pallet placard format:

- 0 – Legacy format. This is the default value.
- 1 – 4 x 7 Intelligent Mail placard.
- 2 – 8.5 x 11 Intelligent Mail placard.

## ptPAUSE\_BEFORE\_FACING\_SLIPS

### Enum Value

7118

#### Data Type

BOOL

#### Description

Determines if the printer will pause before printing the facing slips.

While the printer is paused, the user can change the paper in a selected printer before the facing slips are printed.

The default value is TRUE.

#### ptPAUSE\_BEFORE\_PALLET\_LABELS

#### Enum Value

7121

#### Data Type

BOOL

#### Description

Determines if the printer will pause before printing the pallet labels. This allows the user to change the paper stock in the selected printer before the labels are printed.

The default value is TRUE.

#### ptPAUSE\_BEFORE\_TRAY\_LABELS

#### Enum Value

341

#### Data Type

BOOL

#### Description

Determines if the printer will pause before printing the tray/sack labels.

While the printer is paused, the user can change the paper in a selected printer before the tray/sack labels are printed.

The default value is TRUE.

## ptPERMIT\_SECONDARY\_CITY

### Enum Value

7130

### Data Type

String

### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the city of the Post Office issuing the secondary permit.

The default value is an empty string.

## ptPERMIT\_SECONDARY\_ACCOUNT\_NUMBER

### Enum Value

7265

### Data Type

String

### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This property specifies the USPS payment account number or the Mail Anywhere account number, which is used in addition to the permit number.

The default value is an empty string.

## ptPERMIT\_SECONDARY\_NUMBER

### Enum Value

7129

### Data Type

String

### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the USPS permit number for the secondary permit.

The default value is an empty string.

### ptPERMIT\_SECONDARY\_STATE

#### Enum Value

7131

#### Data Type

String

### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the State of the Post Office issuing the secondary permit.

The default value is an empty string.

### ptPERMIT\_SECONDARY\_ZIP

#### Enum Value

7132

#### Data Type

String

### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the ZIP Code of the Post Office issuing the secondary permit.

The default value is an empty string.

### ptPOSTALONE\_JOB\_ID

#### Enum Value

7267

#### Data Type

String

#### Description

Returns the PostalOne Job ID, which you can use to search on the PostalOne! portal, provide additional information about the mailer on tags, or append to file names.

This property is read-only and can only be retrieved after calling DoSort.

The default is an empty string.

#### ptPERMIT\_ZIPCODE

#### Enum Value

532

#### Data Type

Long

#### Description

The ZIP Code where the permit was issued.

The default is The ZIP Code of the permit holder.

#### ptPREFERRED\_CONTAINER\_TYPE

#### Enum Value

6794

#### Data Type

Long

#### Description

Specifies the preferred container type, if any:

- 0 – No preference. This is the default value.
- 1 – 1- & 2-foot trays.
- 2 – 1-foot trays.

- 3 – 2-foot trays.
- 4 – EMM trays.
- 5 – Tubs.
- 6 – Sacks
- 7 – Carton
- 8 – 1-foot and EMM trays

### ptPRESORT\_SEQUENCING\_DATE\_ECR

#### Enum Value

7022

#### Data Type

String

#### Description

Used to set the "Date of Carrier Route Sequencing" on the postage statement

### ptPRESORT\_TEMPLATE\_LIST

#### Enum Value

605

#### Data Type

String

#### Description

Returns a semicolon-separated string of all currently defined templates in the ini file specified by MRTK global property `mrtkSETTINGS_INI_FILE_NAME`.

The default value is an empty string.

### ptPRINT\_ALL\_REPORTS

#### Enum Value

334

## Data Type

BOOL

## Description

Determines whether all reports should be printed.

The default value is FALSE.

**ptPRINT\_CASS**

## Enum Value

340

## Data Type

BOOL

## Description

Determines if PS Form 3553 should be printed.

The default value is FALSE.

**ptPRINT\_FACING\_SLIPS**

## Enum Value

7110

## Data type

bool

## Description

Determines whether to print facing slips.

Default value is "false".

**ptPRINT\_FILE\_TRAYLABELS**

## Enum Value

345

## Data Type

BOOL

#### Description

Determines if tray/sack labels should be printed to a text file.

The default value is FALSE.

#### ptPRINT\_MANIFEST

#### Enum Value

652

#### Data Type

BOOL

#### Description

Determines if mixed-weight First-Class (Manifesting) report should be printed.

The default value is FALSE.

#### ptPRINT\_PALLET\_LABELS

#### Enum Value

7109

#### Data type

bool

#### Description

Determines whether to print pallet labels.

Default value is "false".

#### ptPRINT\_POSTAGE

#### Enum Value

337

#### Data Type

BOOL

#### Description

Determines if the Postage Form should be printed.

The default value is FALSE.

### ptPRINT\_POSTAGE\_SUMMARY

#### Enum Value

656

#### Data Type

BOOL

#### Description

Determines if mixed-weights postage summary report should be printed.

The default value is FALSE.

### ptPRINT\_PRESORT

#### Enum Value

336

#### Data Type

BOOL

#### Description

Determines if the Presort Report should be printed.

The default value is FALSE.

### ptPRINT\_QUALIFICATION

#### Enum Value

335

#### Data Type

BOOL

#### Description

Determines if the Qualification Report should be printed.

The default value is FALSE.

## ptPRINT\_SETTINGREPORT

### Enum Value

350

### Data Type

BOOL

### Description

Determines if the Settings Report should be printed.

The default value is FALSE.

## ptPRINT\_TRAY\_LABELS\_ONLY

### Enum Value

7123

### Data Type

BOOL

### Description

Determines whether to print tray labels only, without additional documents.

Default value is "false".

Set to "true" to print tray labels only, without additional documents.

## ptPRINT\_TRAYLABELS

### Enum Value

338

### Data Type

BOOL

### Description

Determines if tray/sack labels should be printed.

The default value is FALSE.

## ptPRINT\_ZONE

### Enum Value

339

### Data Type

BOOL

### Description

Determines if the Zone Report should be printed.

This report is printed only if a Periodicals presort is being performed.

The default value is FALSE.

## ptPUB\_CONTACT\_NAME

### Enum Value

680

### Data Type

String

### Description

The contact name for a Periodicals sort.

The default value is an empty string.

## ptPUB\_CONTACT\_PHONE

### Enum Value

681

### Data Type

String

### Description

The contact phone for a Periodicals sort.

The default value is an empty string.

## ptPUB\_EDITION\_ISSUE

### Enum Value

692

### Data Type

String

### Description

Contains either the edition code or issue number for this mailing, both which can be up to six characters.

A volume number plus an issue number constitute an edition code.

You must set this property if you set ptPUB\_VOLUME\_NUMBER.

The default value is an empty string.

## ptPUB\_ENTRY\_STATE\_ZIP4

### Enum Value

683

### Data Type

String

### Description

The entry city, state and ZIP Code for a Periodicals sort.

The default value is an empty string.

## ptPUB\_ISSUE\_DATE

### Enum Value

693

### Data Type

String

### Description

The publication issue date for a Periodicals sort.

The default value is an empty string.

#### ptPUB\_ISSUE\_FREQUENCY

Enum Value

694

Data Type

String

Description

The publication issue frequency for a Periodicals sort.

The default value is an empty string.

#### ptPUB\_OWNER\_AGENT\_NAME

Enum Value

684

Data Type

String

Description

The publication owner name for a Periodicals sort.

The default value is an empty string.

#### ptPUB\_TITLE

Enum Value

691

Data Type

String

Description

The title of the publication.

The default value is an empty string.

## ptPUB\_VOLUME\_NUMBER

### Enum Value

6840

### Data Type

String

### Description

The volume number, which can be up to five characters.

You can specify this instead of an edition code.

A volume number plus an issue number constitute an edition code.

If you set this property, you must also set ptPUB\_EDITION\_ISSUE.

The default value is an empty string.

## ptRECORD\_COUNT

### Enum Value

301

### Data Type

Long

### Description

Total number of records to be processed.

The default value is 0.

## ptRECORD\_COUNT\_PER\_RECEIVE

### Enum Value

310

### Data Type

Long

### Description

The number of records that will be returned from each call to the PresortTask.Retrieve function

The default value is 20.

## ptREPORT\_FILE\_TRAYLABELS

### Enum Value

346

### Data Type

String

### Description

Specifies the file name to use when printing the tray or sack labels.

The default value is an empty string.

## ptREPORT\_FILE\_NAME\_MAILDAT

### Enum Value

7268

### Data Type

String

### Description

Returns the name of the Mail.dat file .zip or files.

File names are alphanumeric 8-character strings. The names, for example, L123abcd, are generated as follows: L123 is the name of your IDEAlliance License Code; abcd is a 4-letter code that is randomly generated for jobs run during 20 second increments.

This property is read-only and can only be retrieved after calling DoSort.

The default is an empty string.

## ptREPORT\_FOLDER\_NAME\_MAILDAT

### Enum Value

365

### Data Type

String

#### Description

The path to the folder in which to store Mail.dat files.

#### ptREPORT\_MAILDAT\_SAVE\_AS\_ZIP

#### Enum Value

7264

#### Data Type

Boolean

#### Description

Specifies whether to save connected Mail.dat files in a zip file.

#### ptREPORT\_PRINTER\_CASS

#### Enum Value

333

#### Data Type

String

#### Description

Specifies the printer to use when printing the Address Correction report (PS Form 3553).

#### ptREPORT\_PRINTER\_DEFAULT

#### Enum Value

327

#### Data Type

String

#### Description

Specifies the default printer to use when printing presort forms.

If set to an empty string (""), then the user's current default printer is used. Otherwise, the name specified by the string is used.

If this property is retrieved after setting one of the report printer values, for example `ptREPORT_PRINTER_CASS`, the value returned will be the empty string ("").

The default value is an empty string.

#### `ptREPORT_PRINTER_MANIFEST`

##### Enum Value

651

##### Data Type

String

##### Description

Specifies the printer to use when printing the mixed-weight First-Class (Manifesting) report.

The default value is an empty string.

#### `ptREPORT_PRINTER_POSTAGE`

##### Enum Value

331

##### Data Type

String

##### Description

The name of the postal presort report for the type of presort chosen. For example, for a First Class postal presort, this property would be set to PS form 3600.

The default value is an empty string.

#### `ptREPORT_PRINTER_PRESORT`

##### Enum Value

329

##### Data Type

String

##### Description

Specifies the printer to use when printing the Presort Report.

The default value is an empty string.

## ptREPORT\_PRINTER\_QUALIFICATION

### Enum Value

328

### Data Type

String

### Description

Specifies the printer to use when printing the Qualification Report.

The default value is an empty string.

## ptREPORT\_PRINTER\_SETTINGREPORT

### Enum Value

351

### Data Type

String

### Description

Specifies the printer to use when printing the Settings Report.

The default value is an empty string.

## ptREPORT\_PRINTER\_TRAYLABELS

### Enum Value

330

### Data Type

String

### Description

Specifies the printer to use when printing tray or sack labels.

The default value is an empty string.

## ptREPORT\_PRINTER\_ZONE

### Enum Value

332

### Data Type

String

### Description

Specifies the printer to use when printing the Zone Report.

The default value is an empty string.

## ptREPORT\_SAVE\_MAILDAT

### Enum Value

364

### Data Type

BOOL

### Description

Determines whether to save the Mail.dat files.

Mail.dat files require an add-on for BCC Architect.

You need to call `PrintPresortReports`, `PreviewPresortReports` or `SavePresortReportsAsPDF` to actually create the Mail.dat files.

The default value is FALSE.

## ptREPORT\_WIZARD\_CAPTION

### Enum Value

306

### Data Type

String

### Description

This property is obsolete.

## ptRIDEALONG\_WEIGHT

### Enum Value

695

### Data Type

FLOAT

### Description

The ride-along weight for a Periodicals sort.

The default value is 0.

## ptSHOW\_PAGE\_SETUP

### Enum Value

325

### Data Type

BOOL

### Description

Not currently implemented.

## ptSAVE\_SHIPPING\_SERVICES\_FILE

### Enum Value

7135

### Data Type

BOOL

### Description

In/Out.

Indicates whether to generate the Shipping Services file.

- 0=false
- 1=true

The default value is 0.

#### ptSHIPPING\_SERVICES\_FILE\_PATH

Enum Value

7136

Data Type

String

Description

In/Out.

The Shipping Services file path and name.

The default value is an empty string.

#### ptSHIPPING\_SERVICES\_LOGIN\_ID

Enum Value

7137

Data Type

String

Description

In/Out.

The Postal Service Logon ID: the logon ID that you use to sign in the USPS FTP site to which you submit your shipping services files. For more information about submitting files via FTP, see [USPS Publication 199](#).

Alphanumeric; maximum 20 characters.

Required if eVS / Shipping Services manifest files are to be produced.

The default value is an empty string.

#### ptSHIPPING\_SERVICES\_SEQUENCE\_NUMBER

Enum Value

7138

#### Data Type

String

#### Description

The sequence number for shipping services.

Optional. Use this field as a way to differentiate shipping services files if you are submitting multiple files to USPS on the same day.

This alphanumeric field can be up to 4 letters or numbers in length, and is appended to the automatically generated filename.

The default value is an empty string.

### ptSHOW\_PRINT\_DIALOG

#### Enum Value

326

#### Data Type

BOOL

#### Description

Determines whether to show the standard Print dialog box when the reports are printed.

The default value is FALSE.

### ptSHOW\_SORT\_PROGRESS

#### Enum Value

303

#### Data Type

BOOL

#### Description

Determines whether to show the Presort Progress dialog box.

The default value is TRUE.

## ptSILENT\_MODE

### Enum Value

206

### Data Type

BOOL

### Description

Determines whether to run the task in silent mode, which disables all dialog boxes, including error messages.

The default value is FALSE.

## ptSIMPLIFIED\_DELIVERY\_STATS\_DATE

### Enum Value

7040

### Data Type

### Description

Indicates the date that this simplified address mailing list was created.

This date will be printed on the postage forms and included in the Mail.dat files and the Mail.XML submission.

## ptSORT\_RESULTS\_FLAG

### Enum Value

302

### Data Type

Long

### Description

Determines how records will be returned after performing presort:

- -1 – Let the user choose whether or not to return the list in presorted order.
- 0 – Presort, but do not return the records in presorted order. This is the default value.

- 1 – Presort and return records in presorted order.
- 2 – Presort and return records in presorted order, but do not offer the Label Wizard.

### ptSORT\_WIZARD\_CAPTION

#### Enum Value

305

#### Data Type

String

#### Description

Sets the caption that displays on the top of Presort Wizard dialog boxes.

The default value is "Presort Wizard".

### ptSTATEMENT\_NUMBER

#### Enum Value

653

#### Data Type

String

#### Description

The statement number for this mailing.

The default value is an empty string.

### ptSUPPRESS\_CARRT\_BASIC\_RATES

#### Enum Value

7054

#### Data Type

BOOL

### Description

When sorting for Carrier Route Saturation or High Density price, mail pieces that do not qualify because of carrier route density requirements will go as Automation price instead of Carrier Route Basic price. It is usually less expensive to qualify pieces at the Automation prices when they do not qualify for Saturation or High Density.

The default value is TRUE

### ptSUPPRESS\_INCOUNTY\_RATES

#### Enum Value

6745

#### Data Type

BOOL

### Description

Determines whether to treat all pieces as outside-county.

The default value is FALSE.

### ptTEMPLATE\_NAME\_TO\_USE

#### Enum Value

151

#### Data Type

String

### Description

The name of the presort template to use.

The default value is an empty string.

### ptUSE\_ACS\_NONE\_FOR\_SAMPLE\_COPIES

#### Enum Value

7274

#### Data Type

Boolean

## Description

Indicates if Periodical sample (non-subscriber) copies use the No Corrections service types (STIDs) for Intelligent Mail barcodes (IMb).

The default value for this property is False.

Set this property to True to apply the Alternative Address Sample Copies No Corrections STIDs to all sample (non-subscriber) copies in a mailing. This means that you are opting not to receive address correction notifications for these pieces, which can result in postage savings. For more information, see [Service Type Identifier \(STID\) Table for Periodicals with Alternative Addressing](#).

## ptUSE\_DETACHED\_ADDRESS\_LABEL

### Enum Value

### Data Type

String

### Description

Specifies what type of Detached Address Label to produce, if any.

- 0 – No detached address label
- 1 – Create standard Detached Address Labels
- 3 – Create Detached Marketing Labels

## ptUSE\_EASYTRACK

### Enum Value

7133

### Data Type

BOOL

### Description

Specifies that the user has chosen to use Track N Trace for IMb Tracing.

Defaults to TRUE if the user has Track N Trace available. Requires Track N Trace add-on serial number.

**NOTE** Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings. It replaces the EasyTrack service. This property now provides access to Track N Trace.

When Track N Trace is used, sequence numbers and Mailer ID used in the IM Barcode is generated via the service and the Mailer ID and any sequence numbers set by the client are ignored. The client property used to set the sequence method `ptIM_SEQUENCING_METHOD` is also ignored because the Track N Trace server manages sequence numbers when tracking is enabled.

Track N Trace by default uses BCC Architect's Informed Visibility Mail Tracking & Reporting (IV-MTR) compatible Mailer ID, but optionally you can register other mail provider's Mailer IDs by contacting BCC Software and registering. It is recommended that if you register a Mailer ID with BCC Software that will only be used for mailings that are submitted with Track N Trace. Using other Mailer IDs also requires approval via the USPS Informed Visibility office.

<https://gateway.usps.com/eAdmin/view/signin>

Sign in to the Business Customer Gateway portal, and then log on to the IV-MTR service.

Track N Trace requires that an initial BCC Software Data Services account is created. This account creation can be done via the Presort Wizard.

User ID and Password for the Track N Trace Web Portal is obtained during initial BCC Architect Data Services account setup or if lost by emailing [BCC Software Technical Support](#).

The Presort Wizard allows the designation of additional fields that can be used for searching on the Track N Trace Web Portal. These include Business name, Contact name, and a Custom Field (`FLD_IM_PIECE_IDENTIFIER`).

Track N Trace reporting and tools can be accessed via the Track N Trace Web Portal: <https://Bc-c.trackntrace.com> ↗.

## ptUSE\_EMM\_TRAYS

### Enum Value

697

### Data Type

BOOL

### Description

Determines whether to use extended managed mail (EMM) trays.

The default value is FALSE.

## ptUSE\_EXCEPTIONAL\_DISPATCH

### Enum Value

6738

### Data Type

BOOL

### Description

Determines whether to use exceptional dispatch, which allows Periodicals mailers to transport pieces directly to destination facilities to improve delivery time.

The default value is FALSE.

## ptUSE\_FIRM\_BUNDLES

### Enum Value

6723

### Data Type

BOOL

### Description

Determines whether to sort multiple pieces addressed to the same location into a firm bundle.

Specify the minimum number of pieces per firm bundle with ptFIRM\_BUNDLE\_MIN.

This property requires an add-on to BCC Architect to function.

Firm bundles only apply to Periodicals mailings.

The default value is FALSE.

## ptUSE\_MAILDAT

### Enum Value

6760

### Data Type

BOOL

## Description

Determines whether to add Mail.dat information to the report XML.

To actually generate the Mail.dat files, you need to:

1. Set the `ptREPORT_FOLDER_NAME_MAILDAT` property.
2. Set `ptREPORT_SAVE_MAILDAT` to `TRUE`.
3. Call `PrintPresortReports` or call `PrintReport` using the `MRTKReportID` enum member `prREPORT_FILE_MAILDAT`.

The default value is `FALSE`.

## `ptUSE_REPOSITIONABLE_NOTES`

### Enum Value

6793

### Data Type

BOOL

## Description

USPS® no longer supports repositionable notes. This property no longer supported.

Indicates if repositionable notes are attached to your mail pieces.

The default value is `FALSE`.

## `ptUSE_SIMPLIFIED_ADDRESSING_PIECE_MAX`

### Enum Value

6815

### Data Type

BOOL

## Description

Determines whether to use the suggested maximum number of mail pieces per bundle in a simplified addressing mailing.

Set this to `TRUE` to use the maximum, `FALSE` to disable it.

The default value is TRUE.

## ptUSE\_USPS\_PROMOTION

### Enum Value

7073

### Data Type

INT

### Description

Specifies whether to use a USPS promotion. Use comma-delimited values to apply multiple promotions, when more than one promotion is available for a given date range. The default value is -1.

Value	Meaning
31	Enable the "Personalized Color Transpromo" promotion, valid from 7/1/2020 – 12/31/2020.
32	Enable the "Emerging and Advanced Technology" promotion, valid from 3/1/2020 – 8/31/2020.
33	Enable the "Informed Delivery" promotion, valid from 9/1/2020 – 11/30/2020.
34	Enable the "Tactile, Sensory & Interactive Mailpiece Engagement" promotion, valid from 2/1/2020 – 7/31/2020.
35	Enable the "Mobile Shopping" promotion, valid from 8/1/2020 – 12/31/2020.
-1	No promotion applied. Use this value to unset the promotion if it has been set incorrectly on a template.

## Presort Wizard Properties

### pwBUNDLE\_PIECE\_MAX

#### Enum Value

7082

#### Data Type

Integer

## Description

Specifies the maximum number of pieces per bundle.

`pwCARTON_PIECE_MAX`

## Enum Value

7079

## Data Type

Integer

## Description

Specifies the maximum number of pieces per carton.

`pwCARTON_WEIGHT_MAX`

## Enum Value

7081

## Data Type

Integer

## Description

Specifies the maximum weight per carton, in pounds.

`pwCREATE_CREATE_FSF_SACKS`

## Enum Value

7277

## Data Type

BOOL

## Description

This property enables the option to be able to create FSF sacks. The property applies to machinable Standard Mail Flats, Periodical Flats, and Bound Printed Matter Flats mail piece types.

## pwCREATE\_LOW\_VOLUME\_PALLETS

### Enum Value

7057

### Data Type

BOOL

### Description

This property will create low volume pallets. This optional sorting preparation will result in the minimum requirement for trays and sacks being overridden for one pallet per PVDS destination in accordance with USPS customer support ruling PS-327. <http://pe.usps.gov/text/csr/ps-327.htm>.

## pwCREATE\_ORIGIN\_DESTINATION\_CONTAINERS

### Enum Value

7058

### Data Type

BOOL

### Description

This property if set to FALSE will skip the optional preparation of destination Origin trays/sacks for PVDS (Plant Verified Drop Shipment) destinations.

The default value is TRUE.

## pwPRESORT\_ADC\_ZIP

### Enum Value

### Data Type

String

### Description

List of ZIP Codes for the current mailing, delimited by commas. The ZIP Code for each ADC facility is either 3-Digit Zip Code. The default value is empty.

## pwPRESORT\_BMC\_ZIP

Enum Value

626

Data Type

String

Description

Not currently implemented.

## pwPRESORT\_CLASS

Enum Value

575

Data Type

Long

Description

Not currently implemented.

## pwPRESORT\_COUNTY\_NAME

Enum Value

585

Data Type

String

Description

Not currently implemented.

## pwPRESORT\_DDU\_ZIP

Enum Value

588

Data Type

String

## Description

List of Destination Entry Unit ZIP Codes for the current mailing, delimited by commas. The ZIP Code for each DDU is either a 5- or 9-Digit Zip Code. The default value is empty.

It is only necessary to specify a 9-digit ZIP Code if it is required to distinguish one DDU from another.

Carrier Route data in the Carrier Route field is required for DDU sorting and rates. Density and sequencing requirements still apply for Carrier Route rate eligibility.

## pwPRESORT\_DEFAULT\_MIN\_WEIGHT

### Enum Value

629

### Data Type

Long

### Description

Not currently implemented.

## pwPRESORT\_ENTRY\_POINT\_BMC

### Enum Value

587

### Data Type

Long

### Description

Not currently implemented.

## pwPRESORT\_ENTRY\_POINT\_NDC

### Enum Value

587

### Data Type

Long

## Description

Not currently implemented.

`pwPRESORT_ENTRY_ZIP_CODE`

## Enum Value

581

## Data Type

Long

## Description

`pwPRESORT_INTELLIGENT_MAIL_ONLY`

## Enum Value

## Data Type

BOOL

## Description

Determines whether BCC Architect uses Intelligent Mail barcodes only.

Once you set this property, it will apply to all sorts for all templates.

`pwPRESORT_MAKE_ONLY_FULL_5_DIGIT_CARRIER_ROUTE_TRAYS`

## Enum Value

655

## Data Type

BOOL

## Description

Determines whether to sort to a 5-digit carrier route tray only if tray is full.

The default value is FALSE.

## pwPRESORT\_MAX\_PER\_TRAY

Enum Value

579

Data Type

Long

Description

Not currently implemented.

## pwPRESORT\_MIN\_PER\_TRAY

Enum Value

580

Data Type

Long

Description

Not currently implemented.

## pwPRESORT\_MIXED\_WEIGHT\_SORT

Enum Value

650

Data Type

BOOL

Description

Set to TRUE for a mixed-weight First-Class sort (manifesting).

The default value is FALSE.

## pwPRESORT\_MULTIPLE\_ENTRY\_POINT

Enum Value

586

Data Type

Long

Description

Not currently implemented

**pwPRESORT\_NDC\_ZIP**

Enum Value

699

Data Type

BOOL

Description

Not currently implemented.

**pwPRESORT\_NONMACHINABLE**

Enum Value

699

Data Type

BOOL

Description

Not currently implemented.

**pwPRESORT\_OPTION**

Enum Value

578

Data Type

Long

Description

Not currently implemented.

## pwPRESORT\_ORIGIN\_KEY

### Enum Value

7269

### Data Type

String

### Description

Indicates the Locale key for the facility that you want to use for a mailing under the USPS Seamless Acceptance program.

The default value is an empty string.

## pwPRESORT\_PIECES\_ARE\_BARCODED

### Enum Value

6751

### Data Type

BOOL

### Description

Indicates whether pieces bear a barcode.

The default value is FALSE.

## pwPRESORT\_PIECES\_INCH

### Enum Value

582

### Data Type

Long

### Description

Returns the number of pieces per inch.

This property is output only and cannot be set.

The default value is 0.

## pwPRESORT\_PIECE\_THICKNESS

### Enum Value

597

### Data Type

Long

### Description

Sets the thickness of all mail pieces.

The default value is 0.

**NOTE** This property is not implemented. Use the pwPRESORT\_PIECES\_INCH property instead.

## pwPRESORT\_RATE

### Enum Value

576

### Data Type

String

### Description

Not currently implemented.

## pwPRESORT\_REDUCE\_OVERFLOW\_CONTAINERS

### Enum Value

6750

### Data Type

BOOL

### Description

Not currently implemented.

## pwPRESORT\_SCF\_ZIP

Enum Value

618

Data Type

String

Description

Not currently implemented.

## pwPRESORT\_TYPE

Enum Value

577

Data Type

Long

Description

Not currently implemented.

## pwPRESORT\_WEIGHT\_PER\_PIECE

Enum Value

583

Data Type

Long

Description

The weight of a mail piece in ounces.

The default value is 0.

## pwPRESORT\_WS\_SORT

Enum Value

584

Data Type

BOOL

Description

Not currently implemented.

**pwSUMMARY\_PIECE**

Enum Value

591

Data Type

String

Description

Not currently implemented.

**pwSUMMARY\_REPORTS**

Enum Value

592

Data Type

String

Description

Not currently implemented.

**pwSUMMARY\_SORT**

Enum Value

590

Data Type

String

Description

Not currently implemented.

## Report Wizard Properties

### rwAGENT\_ADDRESS\_ADDR

#### Enum Value

511

#### Data Type

String

#### Description

The address of the mailing agent.

This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.

The default value is an empty string.

### rwAGENT\_ADDRESS\_CITY

#### Enum Value

512

#### Data Type

String

#### Description

The city of the mailing agent.

This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.

The default value is an empty string.

### rwAGENT\_ADDRESS\_CONTACT

#### Enum Value

515

#### Data Type

String

#### Description

The contact name of the mailing agent.

This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.

The default value is an empty string.

#### rwAGENT\_ADDRESS\_EMAIL

##### Enum Value

516

##### Data Type

String

#### Description

The e-mail address of the mailing agent.

This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.

The default value is an empty string.

#### rwAGENT\_ADDRESS\_NAME

##### Enum Value

510

##### Data Type

String

#### Description

The company name of the mailing agent preparing the mailing. A mailing agent can prepare a mailing on behalf of an organization.

This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.

The default value is an empty string.

## rwAGENT\_ADDRESS\_PHONE

### Enum Value

517

### Data Type

String

### Description

The phone number of the mailing agent.

The default value is an empty string.

## rwAGENT\_ADDRESS\_STATE

### Enum Value

513

### Data Type

String

### Description

The state for the mailing agent.

The default value is an empty string.

## rwAGENT\_ADDRESS\_ZIP

### Enum Value

514

### Data Type

String

### Description

The ZIP Code for the mailing agent.

The default value is an empty string.

## rwMAILING\_DATE

### Enum Value

528

### Data Type

String

### Description

The date of that this mailing will be delivered to the USPS.

The default value is the date that this list was last presorted.

## rwNEWSPAPER\_LABELS

### Enum Value

564

### Data Type

BOOL

### Description

Not currently implemented.

## rwNON\_ADVERTISE\_AMOUNT

### Enum Value

526

### Data Type

Long

### Description

The percentage of non-advertising content for a Periodicals mail piece.

The default value is 0.

## rwORG\_ADDRESS\_ADDR

### Enum Value

521

#### Data Type

String

#### Description

The address of the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

#### rwORG\_ADDRESS\_CITY

#### Enum Value

522

#### Data Type

String

#### Description

The city for the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

#### rwORG\_ADDRESS\_CONTACT

#### Enum Value

535

#### Data Type

String

#### Description

The contact name of the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

## rwORG\_ADDRESS\_EMAIL

### Enum Value

536

### Data Type

String

### Description

The e-mail address of the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

## rwORG\_ADDRESS\_NAME

### Enum Value

520

### Data Type

String

### Description

The company name of the organization on whose behalf a mailing is being prepared.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

## rwORG\_ADDRESS\_PHONE

### Enum Value

537

### Data Type

String

### Description

The phone number of the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

#### rwORG\_ADDRESS\_STATE

##### Enum Value

523

##### Data Type

String

##### Description

The state for the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

#### rwORG\_ADDRESS\_ZIP

##### Enum Value

524

##### Data Type

String

##### Description

The ZIP Code of the mailing organization.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value is an empty string.

#### rwORG\_MAILER\_ID

##### Enum Value

539

**Data Type**

String

**Description**

This property no longer has any effect. Use ptMAIL\_OWNER\_MAILER\_ID or ptMAILING\_AGENT\_MAILER\_ID instead.

The default value is an empty string.

**rwORG\_NONPROFIT\_AUTH\_NO****Enum Value**

538

**Data Type**

String

**Description**

Specifies the permit holder's nonprofit authorization number.

The default value is an empty string.

**rwPAYMENT\_ACCOUNT\_NUMBER****Enum Value**

7178

**Data Type**

String

**Description**

Sets the:

- Mail Anywhere account number

-or-

- EPS account number

Accepts up to 20 alphanumeric characters.

Default value is empty.

**NOTE** You may have set a CAPS account number if you set up your permit before the CAPS system was retired. Update this account number to your EPS ID.

## rwPAYMENT\_OPTION

### Enum Value

7240

### Data Type

String

### Description

A string that specifies the Mail.dat MPA postage payment option:

0 – CPP

1 – PVDS

2 – EPS

3 – Debit

4 – Billing

5 – Other

The default value is CPP for periodicals, and Other for other mailing types. Which option you can specify depends on the permit type that you are using.

**NOTE** Option 2 was changed from CAPS to EPS in the August – September 2019 Service Pack 1.

## rwPAYMENT\_TYPE

### Enum Value

531

### Data Type

Long

### Description

Not currently implemented.

## rwPERMIT\_ADDRESS\_ADDR

### Enum Value

501

### Data Type

String

### Description

Address of permit holder.

The default value is an empty string.

## rwPERMIT\_ADDRESS\_CITY

### Enum Value

502

### Data Type

String

### Description

The city of the permit holder.

The default value is an empty string.

## rwPERMIT\_ADDRESS\_CONTACT

### Enum Value

505

### Data Type

String

### Description

The contact name of the permit holder.

The default value is an empty string.

## rwPERMIT\_ADDRESS\_EMAIL

### Enum Value

506

### Data Type

String

### Description

The e-mail of the permit holder.

The default value is an empty string.

## rwPERMIT\_ADDRESS\_NAME

### Enum Value

500

### Data Type

String

### Description

The company name of the permit holder.

The default value is an empty string.

## rwPERMIT\_ADDRESS\_STATE

### Enum Value

503

### Data Type

String

### Description

The state of permit holder.

The default value is an empty string.

## rwPERMIT\_ADDRESS\_ZIP

### Enum Value

504

### Data Type

String

### Description

The ZIP Code of the permit holder.

The default value is an empty string.

## rwPERMIT\_CAPS\_CUSTOMER\_ID

### Enum Value

6705

### Data Type

String

### Description

Deprecated. The permit holder's ID for the Centralized Account Processing System (CAPS) electronic payment system.

The default value is an empty string.

If used, the value in this field is combined with that in the `rwPrint_PRESORT_MAILER_ID` field and can be used to identify a mailing on the *PostalOne!* dashboard.

## rwPERMIT\_MAILER\_ID

### Enum Value

508

### Data Type

String

### Description

This property no longer has any effect. Use `ptMAIL_OWNER_MAILER_ID` or `ptMAILING_AGENT_MAILER_ID` instead.

The default value is an empty string.

#### rwPERMIT\_NONPROFIT\_AUTH\_NO

##### Enum Value

507

##### Data Type

String

##### Description

Specifies the permit holder's nonprofit authorization number.

The default value is an empty string.

#### rwPERMIT\_NUMBER

##### Enum Value

529

##### Data Type

String

##### Description

The permit number for the permit holder.

The default value is an empty string.

This field is limited to 8 alphanumeric characters.

#### rwPOST\_OFFICE

##### Enum Value

527

##### Data Type

String

##### Description

The Post Office where this mailing will be deposited.

The default value is an empty string.

## rwPOSTAGE\_AMOUNT

### Enum Value

525

### Data Type

Long

### Description

The minimum postage affixed to each piece when the payment method is meter minimum or stamp.

The default value is an empty string.

## rwPRINT\_PRESORT\_CITYSTATE

### Enum Value

542

### Data Type

String

### Description

The mailer's city and state that will be printed on tray or sack labels.

The default value is an empty string.

## rwPRINT\_PRESORT\_MAILER\_ID

### Enum Value

544

### Data Type

String

### Description

The Job ID that will be printed on the Qualification Report.

The default value is an empty string.

If used, the value in this field is combined with that in the `rwPERMIT_CAPS_CUSTOMER_ID` and can be used to identify a mailing on the *PostalOne!* dashboard.

#### `rwPRINT_PRESORT_NAME`

Enum Value

540

Data Type

String

Description

The mailer's company name that will be printed on tray or sack labels.

The default value is an empty string.

#### `rwPRINT_PRESORT_PUB_ID`

Enum Value

543

Data Type

String

Description

The publication ID number that will be printed on Qualification Report for a Periodicals mailing.

The default value is an empty string.

#### `rwPRINT_PRESORT_ZIPCODE`

Enum Value

541

Data Type

String

Description

The mailer's ZIP Code that will be printed on tray or sack labels.

The default value is an empty string.

## rwSACK\_LABEL\_COLUMNS

### Enum Value

561

### Data Type

Long

### Description

The number of sack labels that will printed across per row.

The default value is two.

## rwSACK\_LABEL\_CONTINUOUS

### Enum Value

557

### Data Type

BOOL

### Description

Determines if sack labels are printed on continuous labels. This usually applies to dot-matrix printers.

The default value FALSE.

## rwSACK\_LABEL\_HEIGHT

### Enum Value

559

### Data Type

Float

### Description

The height of each sack label in inches if printing to a sheet.

This property is not set if printing to continuous paper.

The default value is 1 inch or 0 if printing to continuous paper.

## rwSACK\_LABEL\_LEFT\_MARGIN

### Enum Value

563

### Data Type

Float

### Description

The left margin in inches when printing sack labels.

The default value is 0.5 inches.

## rwSACK\_LABEL\_ROWS

### Enum Value

560

### Data Type

Long

### Description

The number of rows of labels.

Set this to 10 or less for sheets, 11 for continuous feed labels.

## rwSACK\_LABEL\_TOP\_MARGIN

### Enum Value

562

### Data Type

Float

### Description

The top margin in inches when printing sack labels.

The default value is 0.5 inches, 0 if printing to continuous paper.

## rwSACK\_LABEL\_WIDTH

### Enum Value

558

### Data Type

Float

### Description

The width of each printed sack label in inches.

The default value is 3.25.

## rwTELEPHONE

### Enum Value

530

### Data Type

String

### Description

The telephone number of the permit holder.

The default value is an empty string.

## rwTRAY\_LABEL\_COLUMNS

### Enum Value

554

### Data Type

Long

### Description

The number of tray labels to be printed across each row.

The default value is two.

## rwTRAY\_LABEL\_CONTINUOUS

### Enum Value

550

### Data Type

BOOL

### Description

Determines if tray labels are printed on continuous labels, such as on a dot-matrix printer.

The default value is FALSE.

## rwTRAY\_LABEL\_HEIGHT

### Enum Value

552

### Data Type

Float

### Description

The height of each tray label in inches if printing to a sheet.

This property is not set if printing to continuous paper.

The default value is 2 or 0 if printing to continuous paper.

## rwTRAY\_LABEL\_LEFT\_MARGIN

### Enum Value

556

### Data Type

Float

### Description

The left margin in inches when printing tray labels.

The default value is 0.5 inches.

## rwTRAY\_LABEL\_ROWS

### Enum Value

553

### Data Type

Long

### Description

The number of rows of labels to print per page.

Set this to 5 or less for sheets, 7 for continuous feed paper.

## rwTRAY\_LABEL\_TOP\_MARGIN

### Enum Value

555

### Data Type

Float

### Description

The top margin in inches when printing tray labels.

The default value is 0.5 inches, 0 if printing to continuous paper.

## rwTRAY\_LABEL\_WIDTH

### Enum Value

551

### Data Type

Float

### Description

The width of each printed tray label in inches.

The default value is 3.25.

## COM PresortTask Properties Summary Table

PresortTask Properties	Enum Value	Data Type	Default Value	Description
------------------------	------------	-----------	---------------	-------------

ptACS_METHOD	7061	Long	0	<p data-bbox="1036 226 1474 296">Determines which service type code is used in the IM Barcode.</p> <ul style="list-style-type: none"> <li data-bbox="1101 338 1422 407">• 0 – None. This is the default value.</li> <li data-bbox="1101 459 1427 487">• 1 – ManualCorrection</li> <li data-bbox="1101 539 1247 567">• 2 – ASR</li> <li data-bbox="1101 619 1247 646">• 3 – CSR</li> <li data-bbox="1101 699 1398 768">• 4 – PeriodicalASRwithASE</li> <li data-bbox="1101 821 1386 848">• 5 – PeriodicalACS</li> <li data-bbox="1101 900 1406 970">• 6 – TraditionalASROption1</li> <li data-bbox="1101 1022 1406 1092">• 7 – TraditionalCSROption1</li> <li data-bbox="1101 1144 1414 1213">• 8 – TraditionalASROption2</li> <li data-bbox="1101 1266 1414 1335">• 9 – TraditionalCSROption2</li> <li data-bbox="1101 1388 1438 1457">• 10 – TraditionalPeriodicalACS</li> <li data-bbox="1101 1509 1362 1537">• 11 – ASROption2</li> <li data-bbox="1101 1589 1252 1617">• 12 – RSR</li> <li data-bbox="1101 1669 1273 1696">• 13 – TRSR</li> <li data-bbox="1101 1749 1403 1776">• 14 – TraditionalRSR</li> <li data-bbox="1101 1829 1422 1856">• 15 – TraditionalTRSR</li> <li data-bbox="1101 1908 1370 1936">• 16 – CSROption2</li> <li data-bbox="1101 1988 1305 2016">• 17 – CSR1SD</li> <li data-bbox="1101 2068 1317 2095">• 18 – CSR2SD</li> </ul>
--------------	------	------	---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ptALWAYS_USE_PERIODICALS_FSS_PREP	7166	string	0	<p>When preparing a Periodicals mailing, will always use FSS if available.</p> <ul style="list-style-type: none"> <li>• 0 = do not automatically use FSS for periodicals</li> <li>• 1 = automatically use FSS for periodicals</li> </ul>
ptANNUAL_NONSUBSCRIBER_THRESHOLD_EXCEEDED	7111	Bool	FALSE	<p>Determines whether the in-county rates are applied to periodicals.</p> <p>Applies only to periodicals.</p> <p>Once a periodical has exceeded the annual 10% threshold of mailing to non-subscribers, they can no longer claim in-county rates for those addresses.</p> <p>Default value is "false"</p> <p>Set to "true" to indicate the annual threshold has been exceeded.</p>
ptAPPLY_BREAK_MARK_INDICATOR	7096	String	0	<p>Specify where to apply a break mark indicator for bundles, trays, or pallets.</p> <ul style="list-style-type: none"> <li>• 0 = apply break mark on first piece</li> <li>• 1 = apply break mark on last piece</li> </ul>

ptAPPLY_PARCEL_ SURCHARGE	7064	BOOL	FALSE	<p>This property will trigger application of the parcel surcharge for the current mailing, if package is not barcoded or if package weighs less than two ounces.</p> <p>The property has no effect on classes other than 1st class. For other classes, it is ignored.</p>
ptBREAK_MARK_IND_ BUNDLE	7093	String	#	Specifies break mark indicator to use for bundle breaks
ptBREAK_MARK_IND_ CONTAINER	7094	String	##	Specifies the break mark indicator to use for container breaks
ptBREAK_MARK_IND_ PALLET	7095	String	###	Specifies the break mark to use for pallet breaks
ptCASS_CERTIFY_FIRST	312	BOOL	FALSE	TRUE to perform address correction prior to presorting
ptCASS_PROCESS_DATE_ AUTOMATION	6761	String	EMPTY	Specifies the CASS processing date to print on the postage statement for automation rate pieces.
ptCASS_PROCESS_DATE_ ECR	6762	String	EMPTY	Specifies the CASS processing date to print on the postage statement for ECR rate pieces.
ptCOMBINE_RESIDUAL_ PIECES	7155	BOOL	TRUE	<p>Use this property to combine 1oz and 2oz residual pieces for a blended rate.</p> <p>The default value for this property is True.</p>

ptCONFIRM_TRACKING	7044	Long	0	<p>Specifies that Track N Trace, the BCC Software implementation of USPS Informed Visibility Mail Tracking &amp; Reporting (IV-MTR), should be used for a mailing. See ptCONFIRM_TRACKING for details.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>• 0 – No tracking</li> <li>• 1 – Tracking via IMb Tracking. For Track N Trace tracking, you must also set ptUSE_EASYTRACK to 1.</li> </ul> <p><b>NOTE</b> Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings. It replaces the EasyTrack service. This property now provides access to Track N Trace.</p>
ptCONTAINER_LABEL_LAYOUT	6819	Long	0	<p>Determines whether sack and tray labels will be printed as Legacy or Intelligent Mail labels. Intelligent Mail labels are required to qualify for Full Service discounts and services. Valid values are:</p> <ul style="list-style-type: none"> <li>• 0 – Legacy</li> <li>• 1 – Intelligent Mail</li> </ul>

ptCREATE_COURTESY_PALLETS	7275	BOOL		<p>This property enables the option to be able to create courtesy pallets, which are a special type of user pallet that can be used to place residual bundles in sacks on a pallet for submission of a mailing under the USPS Seamless Acceptance program.</p> <p>Pallet placards for these pallets are marked as Mixed-NDC.</p>
ptCUSTOM_BARCODE_MAIL_ER_ID	7134	String	EMPTY	<p>Third party mailer ID to use for IMb Tracing™.</p> <p>Must be a Mailer ID registered with USPS® for IMb Tracing.</p>
ptDATA_SERVICES_CLENT_ID_LIST	7263	String	EMPTY	<p>Returns a list of client IDs for a specific provider ID that uses BCC Architect Track N Trace. The default value is an empty string.</p>
ptDATA_SERVICES_CLIENT_USER	7059	String	EMPTY	<p>The name of the client (if any) associated with the current sort template.</p>
ptDATA_SERVICES_JOB_ID	7062	String	EMPTY	<p>Specifies the job ID associated with the BCC Architect Track N Trace job.</p>
ptDATA_SERVICES_PASSWORD	7002	String	EMPTY	<p>The BCC Architect Data Services password for the Provider account used with BCC Architect Track N Trace.</p>
ptDATA_SERVICES_USER	7001	String	EMPTY	<p>The user name for the Data Services Provider account with BCC Software Track N Trace.</p>

ptDROP_SHIP_ADC_ZIPS		String	EMPTY	List of ADC facilities for Plant-Verified Drop Shipment, delimited by commas. The format for each facility is the 3-Digit Zip Code, optionally followed by a colon and minimum weight, in pounds. If a minimum weight is not supplied, the default minimum weight will be used.
ptDROP_SHIP_BMC_ZIPS	626	String	EMPTY	Use ptDROP_SHIP_NDC_ZIPS instead. This property is obsolete.
ptDROP_SHIP_NDC_ZIPS				List of NDC facilities for Plant-Verified Drop Shipment, delimited by commas. The format for each facility is the 3- or 5-Digit Zip Code, optionally followed by a colon and minimum weight, in pounds. If a minimum weight is not supplied, the default minimum weight will be used.
ptDROP_SHIP_SCF_ZIPS	618	String	EMPTY	List of SCF facilities for Plant-Verified Drop Shipment, delimited by commas. The format for each facility is the 3-Digit Zip Code, optionally followed by a colon and minimum weight, in pounds. If a minimum weight is not supplied, the default minimum weight will be used.
ptENABLE_CASS_PROCESSING	307	BOOL	TRUE	TRUE to enable printing PS Form 3553 in Presort Wizard
ptENABLE_TEMPLATES	308	BOOL	TRUE	TRUE to display the list of saved Presort templates in the Presort Wizard

ptEXCEPTIONAL_ DISPATCH_ZIPS	6739	String	EMPTY	List of 5-digit ZIP Codes, delimited by commas, that will be used for Periodicals exceptional dispatch.
ptFIRM_BUNDLE_MIN	6721	Long	2	Determines the minimum number of pieces required to make a firm bundle. Enable firm bundles with the ptUSE_FIRM_BUNDLES property. This property requires an add-on to BCC Architect to function. Firm bundles only apply to Periodicals mailings.
ptFORCE_WALK_ SEQUENCE_SATURATION	815	BOOL	FALSE	TRUE to apply saturation rates to all ECR pieces without verifying density requirements.
ptFORM_NAME	300	String	"Mailing List"	Name of the list you are processing
ptHIDE_SORT_PROGRESS_ AFTER_ SORT	304	BOOL	FALSE	TRUE to hide progress dialog after presorting is complete
ptIM_BARCODE_MAILER_ ID_CODE	6831	Long	0	Determines which Mailer ID to use: <ul style="list-style-type: none"> <li>• 0 – List owner as specified by ptMAIL_OWNER_MAILER_ID</li> <li>• 1 – Mailing agent, as specified by ptMAILING_AGENT_MAILER_ID</li> </ul>

ptIM_CONTAINER_SEQUENCE_LAST	6826	Long	0	Contains the last sequence number used for Intelligent Mail barcodes on a tray or sack. You cannot set this property. If no sequence numbers have been used previously, this property will return 0.
ptIM_CONTAINER_SEQUENCE_START	6823	Long	1	The first sequence number for Intelligent Mail barcodes on trays or sacks when you specify manual sequencing in ptIM_SEQUENCING_METHOD.
ptCUSTOM_BARCODE_MAILER_ID	7134	String	EMPTY	Third party mailer ID to use for IMb Tracing™.  Must be a Mailer ID registered with USPS® for IMb Tracing.
ptIM_EINDUCTION	7234	Bool	False	Indicates whether the Intelligent Mail mailing uses eInduction.
ptIM_EINDUCTION_ACCEPT_MISSHIPPED	7235	Bool	False	Indicates whether there is support for misshipped containers in the eInduction mailing.
ptIM_EINDUCTION_FAST_SCHEDULER_ID	7236	String	EMPTY	Specifies the Fast Scheduler ID.
ptIM_PALLET_SEQUENCE_LAST	6827	Long	0	Contains the last sequence number used for Intelligent Mail barcodes on a pallet. You cannot set this property. If no sequence numbers have been used previously, this property will return 0.

ptIM_PALLET_SEQUENCE_START	6824	Long	1	The first sequence number for Intelligent Mail barcodes on pallets when you specify manual sequencing in ptIM_SEQUENCING_METHOD.
ptIM_PIECE_SEQUENCE_LAST	6825	Long	0	Contains the last sequence number used for Intelligent Mail barcodes on a mail piece. You cannot set this property. If no sequence numbers have been used previously, this property will return 0.
ptIM_PIECE_SEQUENCE_START	6822	Long	1	The first sequence number for Intelligent Mail barcodes on mail pieces when you specify manual sequencing in ptIM_SEQUENCING_METHOD.

ptIM_SEQUENCING_METHOD	6821	Long	1	<p>Determines the sequencing method for the Intelligent Mail barcode on mail pieces, containers and pallets. Valid values are:</p> <ul style="list-style-type: none"> <li>• 0 – A single number will be used for all barcodes. You will not qualify for Full Service discounts and services with this value.</li> <li>• 1 – Automatic numbering based on previously used values and valid ranges.</li> <li>• 2 – Manual numbering, beginning with the value specified in ptIM_PIECE_SEQUENCE_START.</li> <li>• 3 – Sequence number will be based on the value in the field, FLD_IM_PIECE_SEQUENCE_ID</li> </ul>
ptINCOUNTY_ZIPS	6746	String	EMPTY	<p>List of 5-digit ZIP Codes, delimited by commas, that will be treated as in-county.</p>
ptJOB_ID	6832	String	EMPTY	<p>Specifies the Job ID, which is printed on the Qualification, Container Listing, and Mail Sort Summary reports.</p> <p>This property should be used in preference to the rwPRINT_PRESORT_MAILER_ID property.</p>

ptLABELS_BARCODE	342	BOOL	FALSE	TRUE if a DP Barcode is required for printing labels after a presort. This property is read only and can only be retrieved after calling DoSort.
ptLABELS_ENDORSEMENT	343	BOOL	FALSE	TRUE if an Endorsement line is required for printing labels after a presort. This property is read only and can only be retrieved after calling DoSort.
ptMAIL_CONTENT	7177	Integer	0	<p>Specifies whether the mailpiece content is eligible for Election Mail or Political Mail discounts.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• 0 – No discount is applied; the mailpiece content does not qualify as Election Mail or Political Mail.</li> <li>• 1 – Official Election Mail</li> <li>• 2 – Political Campaign Mail</li> </ul> <p>Default value is 0.</p>
ptMAIL_OWNER_CRID	6841	String	EMPTY	The Customer Registration ID for the owner of this mailing.

ptMAIL_OWNER_MAILER_ID	6828	String	EMPTY	Contains the Mailer ID for the owner of the mailing, regardless of whether they are sorting and sending it. You can specify multiple Mailer IDs, separated by commas. These additional Mailer IDs will be used if you selected Automatic as the ptIM_SEQUENCING_METHOD, and the first Mailer ID does not have enough unused sequence numbers to cover all mail pieces in the mailing. All other sequencing methods will only use the first Mailer ID.
ptMAILDAT_CONTACT_EMAIL	6757	String	EMPTY	A string of at most 60 characters that specifies the Mail.dat contact e-mail address
ptMAILDAT_CONTACT_NAME	6755	String	EMPTY	An alphanumeric of at most 30 characters that specifies the Mail.dat contact name
ptMAILDAT_CONTACT_PHONE	6756	String	EMPTY	A ten-digit number that specifies the Mail.dat contact phone
ptMAILDAT_INFORMED_CODE	7247	String	EMPTY	Unique code for the Informed Delivery campaign.  Maximum length: 40
ptMAILDAT_INFORMED_END	7253	String	EMPTY	Date the Informed Delivery campaign is to end.  Leave blank when there is no end date. Do not replace with zeroes.

ptMAILDAT_INFORMED_ID	7256	String	Empty	Unique ID individual Informed Delivery campaign. Can be the same ID as the value in the Campaign Code property.
ptMAILDAT_INFORMED_NAME	7248	String	EMPTY	Descriptive name for your mailing campaign. The name is used in the Informed Delivery email message to identify who sent the mailing.
ptMAILDAT_INFORMED_REPRESENT	7251	String	EMPTY	URL pointer to an image to replace the default scanned gray-scale image of the mail piece.  Maximum length: 250
ptMAILDAT_INFORMED_RIDEALONG	7249	String	EMPTY	URL pointer to an image that includes a clear call to action or next steps.  Maximum length: 250
ptMAILDAT_INFORMED_START	7252	String	EMPTY	Date the Informed Delivery campaign is to begin.  Leave blank when there is no start date. Do not replace with zeroes.
ptMAILDAT_INFORMED_TARGETURL	7250	String	EMPTY	URL pointer to a website or portal to provide a customer with more information.  Maximum length: 250
ptMAILDAT_INFORMED_TITLE	7246	String	EMPTY	Descriptive name for your mailing campaign. This is for informational purposes and is visible only to the mailer in the Informed Delivery mailer portal.

ptMAILDAT_MAILING_ FACILITY_ID	6753	String	EMPTY	An alphanumeric of at most 10 characters that specifies the Mail.dat mailing facility ID
ptMAILDAT_MAILING_TITLE	6854	String	EMPTY	A string that contains the mailing title for the Mail.dat files.
ptMAILDAT_PERMIT_ HOLDER_ID	6754	String	EMPTY	A string of at most 8 characters that specifies the Mail.dat permit holder ID.
ptMAILDAT_USER_ LICENSE_CODE	6752	String	EMPTY	A four-character alphanumeric starting with a letter that specifies the Mail.dat user license code
ptMAILDAT_VERIFICATION_ FACILITY_ NAME	6758	String	EMPTY	An alphanumeric of at most 30 characters that specifies the Mail.dat facility name
ptMAILDAT_VERIFICATION_ FACILITY_ ZIP4	6759	String	EMPTY	A nine-digit number that specifies the Mail.dat verification facility ZIP+4 Code
ptMAILDAT_VERSION	6843	String	"18-1"	<p>Sets the version of the Mail.dat files that will be generated.</p> <p>For mailings on or after March 25, 2018, set this to either "17-1" or "18-1".</p> <p>The default value is "18-1".</p> <p><b>IMPORTANT</b> Mail.dat versions can change frequently, check with PostalOne! and BCC Architect Technical Support for more details.</p>

ptMAILER_ID_USED	6830	String	EMPTY	Returns the Mailer ID that was used when multiple Mailer IDs are specified. You cannot set this property.
ptMAILING_AGENT_CRID	6842	String	EMPTY	The Customer Registration ID for the company performing the mailing, if different from the mail owner.
ptMAILING_AGENT_MAILER_ID	6829	String	EMPTY	Contains the Mailer ID for the mailing agent if this mailing is being sent on behalf of another organization. You can specify multiple Mailer IDs, separated by commas. These additional Mailer IDs will be used if you selected Automatic as the ptIM_SEQUENCING_METHOD, and the first Mailer ID does not have enough unused sequence numbers to cover all mail pieces in the mailing. All other sequencing methods will only use the first Mailer ID.
ptMAILING_IS_PLUS_ONE_MARRIAGE_MAIL	7279	BOOL	False	Indicates if the mailing is PlusOne Marriage Mail.
ptMANIFEST_SEQUENCE_NUMBER	7020	String	EMPTY	Used to set the Manifest sequence number on the Manifest report.
ptMOVE_UPDATE_DATE	6925	String	EMPTY	Contains the date that this entire mailing list was last processed. This will be printed on the postal forms.

ptMOVE_UPDATE_METHOD	6806	Long	0	<p>Determines which box is checked in the Move Update Method section of the Postage Statement and saved in the Mail.DAT files:</p> <ul style="list-style-type: none"> <li>• 0 – None</li> <li>• 1 – NCOA</li> <li>• 2 – ACS</li> <li>• 3 – Ancillary Service Endorsement</li> <li>• 4 – Alternative</li> <li>• 6 – Multiple</li> <li>• 7 – OneCode ACS</li> <li>• 8 – n/a Alternative Address Format</li> <li>• 9 – Automatic (checks ptACS_Method property to see if OneCode ACS is used.)</li> </ul> <p><b>IMPORTANT</b> USPS has retired FastForward. The previously available value “5 – FastForward” is no longer valid as a selection.</p>
ptMRTK_PREFER_CONDENSED_REPORTS	6744	BOOL	FALSE	TRUE to print EZ version of PS 3600 and PS 3602 when possible.

ptMRTK_VERSION	212	String	EMPTY	Returns the current BCC Architect server version.
ptOWNER_GHOST_NUMBER	534	String	EMPTY	Identifies the Mail Owner when the Mailing Agent uses a permit other than the Mail Owners' permit.
ptPALLET_PLACARD_LAYOUT	6820	Long	0	Determines the pallet placard format. Valid values are: <ul style="list-style-type: none"> <li>• 0 – Legacy format</li> <li>• 1 – 4 x 7 Intelligent Mail placard</li> <li>• 2 – 8.5 x 11 Intelligent Mail placard</li> </ul>
ptPAUSE_BEFORE_FACING_SLIPS	7118	BOOL	TRUE	TRUE to pause before printing the facing slips. This allows the user to change the paper in a selected printer before the facing slips are printed.
ptPAUSE_BEFORE_PALLET_LABELS	7121	BOOL	TRUE	TRUE if the printer will pause before printing the pallet labels. This allows the user to change the paper stock in the selected printer before the labels are printed.
ptPAUSE_BEFORE_TRAY_LABELS	341	BOOL	TRUE	TRUE if the printer will pause before printing the tray/sack labels. This allows the user to change the paper in a selected printer before the tray/sack labels are printed.

ptPERMIT_SECONDARY_CITY	7130	String	EMPTY	When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the city of the Post Office issuing the secondary permit.
ptPERMIT_SECONDARY_ACCOUNT_NUMBER	7265	String	EMPTY	When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This property specifies the USPS payment account number or the Mail Anywhere account number, which is used in addition to the permit number.
ptPERMIT_SECONDARY_NUMBER	7129	String	EMPTY	When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the USPS permit number for the secondary permit.
ptPERMIT_SECONDARY_STATE	7131	String	EMPTY	When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the State of the Post Office issuing the secondary permit.

ptPERMIT_SECONDARY_ZIP	7132	String	EMPTY	When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the ZIP Code of the Post Office issuing the secondary permit.
ptPERMIT_ZIPCODE	532	Long	See desc.	The ZIP Code where the permit was issued. The default is The ZIP Code of the permit holder.
ptPOSTALONE_JOB_ID	7267	String	EMPTY	Returns the PostalOne Job ID, which you can use to search on the PostalOne! portal, provide additional information about the mailer on tags, or append to file names.
ptPREFERRED_CONTAINER_TYPE	6794	Long	0	Specifies the preferred container type, if any: <ul style="list-style-type: none"> <li>• 0 – No preference</li> <li>• 1 – 1- &amp; 2-foot trays</li> <li>• 2 – 1-foot trays</li> <li>• 3 – 2-foot trays</li> <li>• 4 – EMM trays</li> <li>• 5 – Tubs</li> <li>• 6 – Sacks</li> <li>• 7 – Carton</li> <li>• 8 – 1-foot and EMM trays</li> </ul>

ptPRESORT_SEQUENCING_DATE_ECR	7022	String	EMPTY	Used to set the “Date of Carrier Route Sequencing” on the postage statement
ptPRESORT_TEMPLATE_LIST	605	String	EMPTY	Returns a semicolon-separated string of all currently defined templates in the ini file specified by MRTK global property mrtkSETTINGS_INI_FILE_NAME
ptPRINT_ALL_REPORTS	334	BOOL	FALSE	TRUE if all reports should be printed
ptPRINT_CASS	340	BOOL	FALSE	TRUE if PS Form 3553 should be printed
ptPRINT_FACING_SLIPS	7110	BOOL	FALSE	TRUE to print facing slips
ptPRINT_FILE_TRAYLABELS	345	BOOL	FALSE	TRUE if tray/sack labels should be printed to a text file.
ptPRINT_MANIFEST	652	BOOL	FALSE	TRUE if mixed-weight First-Class (Manifesting) report should be printed
ptPRINT_PALLET_LABELS	7109	BOOL	FALSE	TRUE to print pallet labels.
ptPRINT_POSTAGE	337	BOOL	FALSE	TRUE if Postage Form should be printed
ptPRINT_POSTAGE_SUMMARY	656	BOOL	FALSE	TRUE if mixed-weights postage summary report should be printed
ptPRINT_PRESORT	336	BOOL	FALSE	TRUE if Presort Report should be printed
ptPRINT_QUALIFICATION	335	BOOL	FALSE	TRUE if Qualification Report should be printed

ptPRINT_SETTINGREPORT	350	BOOL	FALSE	TRUE if Settings Report should be printed
ptPRINT_TRAY_LABELS_ONLY	7123	BOOL	FALSE	TRUE to print tray labels only, without additional documents.
ptPRINT_TRAYLABELS	338	BOOL	FALSE	TRUE if tray/sack labels should be printed
ptPRINT_ZONE	339	BOOL	FALSE	TRUE if Zone Report should be printed; this report is printed only if a Periodicals presort is being performed
ptPUB_CONTACT_NAME	680	String	EMPTY	Contact name for a Periodicals sort
ptPUB_CONTACT_PHONE	681	String	EMPTY	Contact phone for a Periodicals sort
ptPUB_EDITION_ISSUE	692	String	EMPTY	Either the edition code or issue number for this mailing, both which can be up to six characters. A volume number plus an issue number constitute an edition code. You must set this property if you set ptPUB_VOLUME_NUMBER.
ptPUB_ENTRY_STATE_ZIP4	683	String	EMPTY	Entry city, state and ZIP Code for a Periodicals sort
ptPUB_ISSUE_DATE	693	String	EMPTY	Publication issue date for a Periodicals sort
ptPUB_ISSUE_FREQUENCY	694	String	EMPTY	Publication issue frequency for a Periodicals sort
ptPUB_OWNER_AGENT_NAME	684	String	EMPTY	Publication owner name for a Periodicals sort

ptPUB_TITLE	691	String	EMPTY	The title of the publication.
ptPUB_VOLUME_NUMBER	6840	String	EMPTY	The volume number, which can be up to five characters. You can specify this instead of an edition code. A volume number plus an issue number constitute an edition code. If you set this property, you must also set ptPUB_EDITION_ISSUE.
ptRECORD_COUNT	301	Long	0	Total number of records to be processed
ptRECORD_COUNT_PER_RECEIVE	310	Long	20	The number of records that will be returned from each call to the PresortTask.Retrieve function
ptREPORT_FILE_TRAYLABELS	346	String	EMPTY	Specifies the file name to use when printing the tray or sack labels
ptREPORT_FILE_NAME_MAILDAT	7268	String	EMPTY	File names are alphanumeric 8-character strings. The names, for example, L123abcd, are generated as follows: L123 is the name of your IDEAlliance License Code; abcd is a 4-letter code that is randomly generated for jobs run during 20 second increments.
ptREPORT_FOLDER_NAME_MAILDAT	365	String	EMPTY	Path to the folder in which to store Mail.dat files
ptREPORT_MAILDAT_SAVE_AS_ZIP	7264	Boolean	FALSE	Specifies whether to save connected Mail.dat files in a zip file.
ptREPORT_PRINTER_CASS	333	String	EMPTY	Specifies the printer to use when printing the CASS report (PS Form 3553)

ptREPORT_PRINTER_DEFAULT	327	String	EMPTY	Specifies the default printer to use when printing presort forms. If set to an empty string (""), then the user's current default printer is used. Otherwise the name specified by the string is used. If this property is retrieved after setting one of the report printer values, for example ptREPORT_PRINTER_CASS, the value returned will be the empty string ("").
ptREPORT_PRINTER_FACING_SLIPS	7117	String	EMPTY	Specifies the printer to use for the facing slips.  The default value for this property is an empty string.
ptREPORT_PRINTER_MANIFEST	651	String	EMPTY	Specifies the printer to use when printing the mixed-weight First-Class (Manifesting) report
ptREPORT_PRINTER_PALLET_LABELS	7120	String	EMPTY	Specifies the printer to use for the pallet labels.  The default value for this property is an empty string.
ptREPORT_PRINTER_POSTAGE	331	String	EMPTY	Depending on the type of presort chosen this property is set with a particular USPS presort report. For example, a First Class postal presort would be set with a PS form 3600.
ptREPORT_PRINTER_PRESORT	329	String	EMPTY	Specifies the printer to use when printing the Presort Report

ptREPORT_PRINTER_QUALIFICATION	328	String	EMPTY	Specifies the printer to use when printing the Qualification Report
ptREPORT_PRINTER_SETTINGREPORT	351	String	EMPTY	Specifies the printer to use when printing the Settings Report
ptREPORT_PRINTER_TRAYLABELS	330	String	EMPTY	Specifies the printer to use when printing tray or sack labels
ptREPORT_PRINTER_ZONE	332	String	EMPTY	Specifies the printer to use when printing the Zone Report
ptREPORT_SAVE_MAILDAT	364	BOOL	FALSE	TRUE to save the Mail.dat files. You need to call PrintPresortReports, PreviewPresortReports or SavePresortReportsAsPDF to actually create the Mail.dat files.
ptREPORT_WIZARD_CAPTION	306	String	"Report Wizard"	This property is obsolete.
ptRIDEALONG_WEIGHT	695	FLOAT	0	Periodicals ride-along weight.
ptSAVE_SHIPPING_SERVICES_FILE	7135	BOOL	FALSE	Indicates whether to generate the Shipping Services file.
ptSHIPPING_SERVICES_FILE_PATH	7136	string	EMPTY	The Shipping Services file path and name.

ptSHIPPING_SERVICES_LOGIN_ID	7137	string	EMPTY	<p>The Postal Service Logon ID: the logon ID that you use to sign in the USPS FTP site to which you submit your shipping services files. For more information about submitting files via FTP, see <a href="#">USPS Publication 199</a>.</p> <p>Alphanumeric; maximum 20 characters.</p> <p>Required if eVS / Shipping Services manifest files are to be produced.</p>
ptSHIPPING_SERVICES_SEQUENCE_NUMBER	7138	string	EMPTY	<p>The sequence number for shipping services.</p> <p>Optional. Use this field as a way to differentiate shipping services files if you are submitting multiple files to USPS on the same day.</p> <p>This alphanumeric field can be up to 4 letters or numbers in length, and is appended to the automatically generated filename.</p>
ptSHOW_PAGE_SETUP	325	BOOL	FALSE	Not currently implemented.
ptSHOW_PRINT_DIALOG	326	BOOL	FALSE	TRUE to show the standard print dialog.
ptSHOW_SORT_PROGRESS	303	BOOL	TRUE	TRUE to show the presort progress dialog.
ptSILENT_MODE	206	BOOL	FALSE	TRUE to run the task in silent mode, which disables all dialogs, including error messages

ptSIMPLIFIED_DELIVERY_STATS_DATE	7040	String	EMPTY	Indicates the date that this simplified address mailing list was created. This date will be printed on the postage forms and included in the Mail.dat files and the Mail.XML submission. You must manually set this date for it to be included in the postal documentation.
ptSORT_RESULTS_FLAG	302	Long	0	Determines how records will be returned after performing pre-sort: <ul style="list-style-type: none"> <li>• -1 = Let the user choose whether or not to return the list in presorted order</li> <li>• 0 = Presort but does not return the records in presorted order</li> <li>• 1 = Presort and return records in presorted order</li> <li>• 2 = Presort and return records in presorted order but do not offer the Label Wizard</li> </ul>
ptSORT_WIZARD_CAPTION	305	String	"Pre-sort Wizard"	Caption that appears on Presort Wizard dialog window
ptSTATEMENT_NUMBER	653	String	EMPTY	The statement number for this mailing.
ptSUPPRESS_INCOUNTY_RATES	6745	BOOL	FALSE	TRUE to treat all pieces as outside-county

ptTEMPLATE_NAME_TO_USE	151	String	EMPTY	The name of the presort template to use for performing the presort
ptUSE_ACS_NONE_FOR_SAMPLE_COPIES	7274	BOOL	FALSE	Indicates if Periodical sample (non-subscriber) copies use the No Corrections service types (STIDs) for Intelligent Mail barcodes (IMb).
ptUSE_DETACHED_ADDRESS_LABEL	7077	Int	0	Specifies what type of Detached Address Label to produce, if any. <ul style="list-style-type: none"> <li>• 0 – No detached address label</li> <li>• 1 – Create standard Detached Address Labels</li> <li>• 3 – Create Detached Marketing Labels</li> </ul>
ptUSE_EASYTRACK	7133	BOOL	TRUE	Specifies that the user has chosen to use Track N Trace for IMb Tracing.  Defaults to TRUE if the user has Track N Trace available.  <b>NOTE</b> Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings. It replaces the EasyTrack service. This property now provides access to Track N Trace.
ptUSE_EMM_TRAYS	697	BOOL	FALSE	TRUE to use extended managed mail (EMM) trays

ptUSE_EXCEPTIONAL_DISPATCH	6738	BOOL	FALSE	TRUE to use exceptional dispatch, which allows Periodicals mailers to transport pieces directly to destination facilities to improve delivery time.
ptUSE_FIRM_BUNDLES	6723	BOOL	FALSE	Set this property to TRUE to sort multiple pieces addressed to the same location into a firm bundle. Specify the minimum number of pieces per firm bundle with ptFIRM_BUNDLE_MIN. This property requires an add-on to BCC Architect to function. Firm bundles only apply to Periodicals mailings.
ptUSE_MAILDAT	6760	BOOL	FALSE	TRUE to add Mail.dat information to the report XML. To actually generate the Mail.dat files, you need to: 1) Set the ptREPORT_FOLDER_NAME_MAILDAT property; 2) Set ptREPORT_SAVE_MAILDAT to TRUE and call PrintPresortReports or call PrintReport using the MRTKReportID enum member prREPORT_FILE_MAILDAT.
ptUSE_REPOSITIONABLE_NOTES	6793	BOOL	FALSE	USPS® no longer supports repositionable notes. This property no longer supported. TRUE if repositionable notes are attached to your mail pieces.

ptUSE_SIMPLIFIED_ ADDRESSING_  PIECE_MAX	6815	BOOL	TRUE	Determines whether to use the suggested maximum number of mail pieces per bundle in a simplified addressing mailing. Set this to TRUE to use the maximum, FALSE to disable it.
---------------------------------------------------	------	------	------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ptUSE_USPS_PROMOTION	7073	int	-1	<p>Specifies whether to use a USPS promotion. Use comma-delimited values to apply multiple promotions, when more than one promotion is available for a given date range.</p> <ul style="list-style-type: none"><li>• 31: Enable the "Personalized Color Transpromo" promotion, valid from 7/1/2020 – 12/31/2020.</li><li>• 32: Enable the "Emerging and Advanced Technology" promotion, valid from 3/1/2020 – 8/31/2020.</li><li>• 33: Enable the "Informed Delivery" promotion, valid from 9/1/2020 – 11/30/2020.</li><li>• 34: Enable the "Tactile, Sensory &amp; Interactive Mailpiece Engagement" promotion, valid from 2/1/2020 – 7/31/2020.</li><li>• 35: Enable the "Mobile Shopping" promotion, valid from 8/1/2020 – 12/31/2020.</li><li>• -1: No promotion applied. Use this value to unset the promotion if it has been set incorrectly on a template.</li></ul>
----------------------	------	-----	----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## COM PresortWizard Properties Summary Table

PresortWizard Properties	Enum Value	Data Type	Default Value	Description
pwBUNDLE_PIECE_MAX	7082	String	EMPTY	The maximum number of pieces per bundle.
pwCARTON_PIECE_MAX	7079	String	EMPTY	
pwCARTON_PIECE_MIN	7078	String	EMPTY	
pwCARTON_WEIGHT_MAX	7081	String	EMPTY	
pwCARTON_WEIGHT_MIN	7080	String	EMPTY	
pwCREATE_CREATE_FSF_SACKS	7277	BOOL	FALSE	This property enables the option to be able to create FSF sacks. The property applies to machinable Standard Mail Flats, Periodical Flats and Bound Printed Matter Flats mail piece types.
pwCREATE_LOW_VOLUME_PALLETS	7057	BOOL		This property will create low volume pallets. This optional sorting preparation will result in the minimum requirement for trays and sacks being overridden for one pallet per PVDS destination in accordance with USPS customer support ruling PS-327. <a href="http://pe.usps.gov/text/csr/ps-327.htm">http://pe.usps.gov/text/csr/ps-327.htm</a> ↗.

PresortWizard Properties	Enum Value	Data Type	Default Value	Description
pwCREATE_ORIGIN_DESTINATION_CONTAINERS	7058	BOOL	TRUE	This property if set to FALSE will skip the optional preparation of destination Origin trays/sacks for PVDS (Plant Verified Drop Shipment) destinations. The default value is TRUE.
pwPRESORT_BMC_ZIP	626	String	EMPTY	Not currently implemented.
pwPRESORT_CLASS	575	Long	0	Not currently implemented.
pwPRESORT_COUNTY_NAME	585	String	EMPTY	Not currently implemented.
pwPRESORT_DDU_ZIP	588	String	EMPTY	List of Destination Entry Unit ZIP Codes for the current mailing, delimited by commas. The ZIP Code for each DUU is either a 5- or 9-Digit Zip Code.
pwPRESORT_DEFAULT_MIN_WEIGHT	629	Long	0	Not currently implemented.
pwPRESORT_ENTRY_POINT_BMC	587	Long	0	Not currently implemented.
pwPRESORT_ENTRY_POINT_NDC	587	Long	0	Not currently implemented.
pwPRESORT_ENTRY_ZIP_CODE	581	Long	0	

PresortWizard Properties	Enum Value	Data Type	Default Value	Description
pwPRESORT_INTELLIGENT_MAIL_ONLY	7039	BOOL	FALSE	Determines whether BCC Architect uses Intelligent Mail barcodes only. Once you set this property, it will apply to all sorts for all templates.
pwPRESORT_MAKE_ONLY_FULL_5_DIGIT_CARRIER_ROUTE_TRAYS	655	BOOL	FALSE	TRUE to sort to a 5-digit carrier route tray only if tray is full.
pwPRESORT_MAX_PER_TRAY	579	Long	0	Not currently implemented.
pwPRESORT_MIN_PER_TRAY	580	Long	0	Not currently implemented.
pwPRESORT_MIXED_WEIGHT_SORT	650	BOOL	FALSE	TRUE for a mixed-weight First-Class sort (manifesting).
pwPRESORT_MULTIPLE_ENTRY_POINT	586	Long	0	Not currently implemented.
pwPRESORT_NDC_ZIP	626	String	EMPTY	Not currently implemented.
pwPRESORT_NONMACHINABLE	699	BOOL	FALSE	Not currently implemented.
pwPRESORT_OPTION	578	Long	0	Not currently implemented.
pwPRESORT_ORIGIN_KEY	7269	String	EMPTY	Indicates the Locale key for the facility that you want to use for a mailing under the USPS Seamless Acceptance program.

PresortWizard Properties	Enum Value	Data Type	Default Value	Description
pwPRESORT_PIECES_ARE_BARCODED	6751	BOOL	FALSE	TRUE if pieces bear a barcode.
pwPRESORT_PIECES_INCH	582	Long	0	Returns the number of pieces per inch. This property is output only and cannot be set.
pwPRESORT_PIECE_THICKNESS	597	Long	0	Sets the thickness of all mail pieces.  <b>NOTE</b> This property is not implemented. Use the pwPRESORT_PIECES_INCH property instead.
pwPRESORT_RATE	576	String	EMPTY	Not currently implemented.
pwPRESORT_REDUCE_OVERFLOW_CONTAINERS	6750	BOOL	FALSE	Not currently implemented.
pwPRESORT_SCF_ZIP	618	String	EMPTY	Not currently implemented.
pwPRESORT_TYPE	577	Long	0	Not currently implemented.
pwPRESORT_WEIGHT_PER_PIECE	583	Long	0	Piece weight in ounces.
pwPRESORT_WS_SORT	584	BOOL	FALSE	Not currently implemented.
pwSUMMARY_PIECE	591	String	EMPTY	Not currently implemented.
pwSUMMARY_REPORTS	592	String	EMPTY	Not currently implemented.
pwSUMMARY_SORT	590	String	EMPTY	Not currently implemented.

## The COM ReviewErrorsTask Object for Reviewing Errors

The ReviewErrorsTask object in BCC Architect enables you to create scripts for viewing and correcting uncorrected or questionable addresses.

Although ReviewErrorsTask is used primarily in conjunction with CASSTask, you can also use it independently allowing for greater flexibility. For example, you can split the list of uncorrected addresses to allow multiple users to work on separate segments. Having an independent ReviewErrorsTask also allows addresses to be edited using our Review Uncorrected Records dialog box at any time. However, addresses corrected independently of the CASSTask will not be reflected in the Address Correction report (PS Form 3553) until the list is reprocessed.

The ReviewErrorsTask should be created via the BCC Architect object factory (MRTKObjFactory) if you use the ReviewErrorsTask independently of CASSTask.

### ReviewErrorsTask Functions

The ReviewErrorsTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

#### PrepareTask

##### Syntax

```
PrepareTask  
  
long PrepareTask()
```

##### Description

Initialize and prepare the object.

##### Parameters

None.

##### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

##### Notes

This function must be called before any of the other functions or properties of ReviewErrorsTask.

If this function is not called first, all subsequent ReviewErrorsTask function calls will fail.

The one exception to this rule is calling SetProperty to set the path of the Address.cas file, which must be defined prior to calling PrepareTask. It is preferable, however, to set the data file path using the MRTKObjFactory.DataFilePath property.

See also

### [COM Factory Object](#)

## GetProperty

Syntax

```
GetProperty(mrtkPropertyID)  
  
long GetProperty(long mrtkPropertyID, VARIANT *pVal)
```

Description

Retrieve a ReviewErrorsTask property value.

Parameters

*mrtkPropertyID as Long*

The ID of the property to retrieve.

*mrtkPropertyID*

The ID of the property to retrieve.

*pVal*

Returns the value of the property.

Return values

The value of the property (as Variant). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

See also

See the [ReviewErrorsTask Properties](#) table for a list of property IDs.

## SetProperty

Syntax

```
SetProperty(mrtkPropertyID, value)
```

```
long SetProperty(long mrtkPropertyID, VARIANT value)
```

#### Description

Set a ReviewErrorsTask property.

#### Parameters

*mrtkPropertyID as Long*

The ID of the property to set.

*value as Variant*

The value of the property to set.

*mrtkPropertyID*

The ID of the property to set.

*value*

The value of the property to set.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### See also

See the [ReviewErrorsTask Properties](#) table for a list of property IDs.

#### ValidateProperties

##### Syntax

```
ValidateProperties
```

```
long ValidateProperties()
```

##### Description

Verify that the task is set up correctly and ready to run.

##### Parameters

None.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function checks to make sure that the address matching engine is loaded and able to run and the input field list consists of the minimum set of fields, namely, Address Line 1 and either City and State or ZIP Code.

This function needs to be called before you call Send.

## See also

See the [ReviewErrorsTask Properties](#) table for a list of property IDs.

## Send

### Syntax

```
Send(strAddressBlock)
```

```
long Send(BSTR *pbstrAddressBlock)
```

### Description

Send a block of addresses to the ReviewErrorsTask object.

### Parameters

*strAddressBlock as String*

A string that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

*pbstrAddressBlock*

A BSTR that contains mrtkINPUT\_BLOCK\_RECORD\_COUNT addresses separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You may want to experiment with the `mrtkINPUT_BLOCK_RECORD_COUNT` property. In preliminary tests, we have found the optimal setting to be around 25-50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500-1000. A setting of 25-50 makes the process run very slowly in client-server mode. Call `ReviewErrors` after you have finished sending all of your records to the `ReviewErrorsTask` object.

If your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix), use the `SendV` function instead.

## See also

- [SendV](#)
- [ReviewErrors](#)

## SendV

### Syntax

```
SendV(strAddressBlock)  
  
long SendV(BSTR *pbstrAddressBlock)
```

### Description

Send a block of addresses to the `ReviewErrorsTask` object.

### Parameters

#### *strAddressBlock as String*

A string that contains `mrtkINPUT_BLOCK_RECORD_COUNT` addresses separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`.

#### *pbstrAddressBlock*

A BSTR that contains `mrtkINPUT_BLOCK_RECORD_COUNT` addresses separated by `mrtkDELIMITER_FIELD` and `mrtkDELIMITER_RECORD`.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You may want to experiment with the `mrtkINPUT_BLOCK_RECORD_COUNT` property. In preliminary tests, we have found the optimal setting to be around 25-50 if you are not using client-server mode.

If you are using client-server mode, we have found the optimal settings to be 500-1000. A setting of 25-50 makes the process run very slowly in client-server mode. Call `ReviewErrors` after you have finished sending all of your records to the `ReviewErrorsTask` object.

You must use this function if your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix). Otherwise, you can use either this function or `Send`.

## See also

- [Send](#)
- [ReviewErrors](#)

## ReviewErrors

### Syntax

```
ReviewErrors  
long ReviewErrors()
```

### Description

Display the Review Uncorrected Addresses dialog box.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

When the Review Uncorrected Addresses dialog box is closed, any addresses that the user keeps will be returned by the `Retrieve` function.

See also

- [Send](#)
- [Retrieve](#)

## Retrieve

Syntax

```
RetrieveReviewed(bstrAddressBlock)  
  
long RetrieveReviewed(BSTR *pbstrAddressBlock)
```

Description

Retrieve the addresses that the user kept in the Review Uncorrected Addresses dialog box.

Parameters

*bstrAddressBlock*

Returns block of corrected addresses that contains etRECORD\_COUNT\_PER\_RECEIVE addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

*pbstrAddressBlock*

Returns block of corrected addresses that contains etRECORD\_COUNT\_PER\_RECEIVE addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

Only those addresses the user chooses to keep in the Review Uncorrected Addresses dialog box will be returned by this function.

If your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix), use the RetrieveV function instead.

See also

[RetrieveV](#)

See the [MRTK Global Properties](#) table for definition of:

- etRECORD\_COUNT\_PER\_RECEIVE
- mrtkDELIMITER\_FIELD
- mrtkDELIMITER\_RECORD
- mrtkFIELD\_LIST\_OUT

## RetrieveV

### Syntax

```
RetrieveReviewedV  
  
long RetrieveReviewedV(BSTR *pbstrAddressBlock)
```

### Description

Retrieve the addresses kept by the user in the Review Uncorrected Addresses dialog box.

### Parameters

None.

*pbstrAddressBlock*

Returns block of corrected addresses that contains etRECORD\_COUNT\_PER\_RECEIVE addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD.

### Return values

Block of corrected addresses that contains etRECORD\_COUNT\_PER\_RECEIVE addresses that are separated by mrtkDELIMITER\_FIELD and mrtkDELIMITER\_RECORD (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Only those addresses the user chooses to keep in the Review Uncorrected Addresses dialog box will be returned by this function.

You must use this function if your development environment does not support passing a variable by reference (e.g., Visual FoxPro or SalesLogix). Otherwise, you can use either this function or Retrieve.

See also

### [Retrieve](#)

See the [MRTK Global Properties](#) table for definition of:

- etRECORD\_COUNT\_PER\_RECEIVE
- mrtkDELIMITER\_FIELD
- mrtkDELIMITER\_RECORD
- mrtkFIELD\_LIST\_OUT

## EndTask

### Syntax

```
EndTask  
  
long EndTask()
```

### Description

When you are finished with the task, clean up and release any resources used.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

Call this function when you have reviewed and retrieved all addresses in the ReviewErrorsTask object.

## ReviewErrorsTask Properties

The ReviewErrorsTask properties shown below are members of the MRTKTASKLib.ReviewErrorsTaskPropertyID enumeration. These enum names are used as arguments for the GetProperty and SetProperty functions. If you have added the BCC Architect files, then you can view all of the available enums in the Object Browser.

**etRECORD\_COUNT\_PER\_RECEIVE**

## Enum Value

375

## Data Type

Long

## Description

Sets and retrieves the number of records that will be returned from each call to the ReviewErrorsTask.Retrieve function.

The default value is 1.

**etSHOW\_RECEIVE\_PROGRESS**

## Enum Value

376

## Data Type

BOOL

## Description

Determines whether to show the Progress dialog box while the client is retrieving the records that are to be saved.

The default value is TRUE.

**COM ReviewErrorsTask Properties Summary Table**

ReviewErrorsTask Properties	Enum Value	Data Type	Default Value	Description
etRECORD_COUNT_PER_RECEIVE	375	Long	1	The number of records that will be returned from each call to the ReviewErrorsTask.Retrieve function
etSHOW_RECEIVE_PROGRESS	376	BOOL	TRUE	TRUE to show a progress dialog while the client is retrieving the "kept" records

## The COM GeocodeTask Object for Adding Geocode Data

The BCC Architect Geocode object returns the latitude and longitude coordinates and related data for a given ZIP Code. It can also compute the distance or angle between two points and apply radius-based filters.

Geocode is a BCC Architect add-on.

### GeocodeTask Functions

The GeocodeTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

#### PrepareTask

##### Syntax

```
PrepareTask  
  
long PrepareTask()
```

##### Description

Initialize and prepare the object.

##### Parameters

None.

##### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

##### Notes

PrepareTask should be called only once, when the GeocodeTask object is first created.

This function must be called before any of the other functions/properties of GeocodeTask. Failing to do so will cause subsequent function calls to fail.

#### AddIsBetweenFilter

##### Syntax

```
AddIsBetweenFilter(bstrZipCode, bstrMinDistance, bstrMaxDistance)
```

```
long AddIsBetweenFilter(BSTR bstrZipCode, BSTR bstrMinDistance, BSTR  
bstrMaxDistance)
```

#### Description

Adds a filter to include points between a minimum distance and maximum distance away from a ZIP Code.

#### Parameters

*bstrZipCode as String*

The ZIP Code whose latitude and longitude coordinates define the origin of the search radius.

*bstrMinDistance as String*

The minimum distance that an address must be to select it, in miles.

*bstrMaxDistance as String*

The maximum distance that an address can be to select it, in miles.

*bstrZipCode*

The ZIP Code whose latitude and longitude coordinates define the origin of the search radius.

*bstrMinDistance*

The minimum distance that an address must be to select it, in miles.

*bstrMaxDistance*

The maximum distance that an address can be to select it, in miles.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This filter will include all points, as defined by their latitude and longitude coordinates, that lie between two concentric circles centered on the point that represents the origin ZIP Code.

This function equivalent to applying the AddRadiusIncludeFilter with bstrMaxDistance radius followed by the AddRadiusExcludeFilter with bstrMinDistance radius.

To use this filter, call either IsPointWithinFilters or IsZipCodeWithinFilters. Those functions will check to see if a point or ZIP Code is within the areas defined by the currently applied filters.

The origin ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will result in more precise coordinates.

See also

[AddRadiusExcludeFilter](#)

[AddRadiusIncludeFilter](#)

[AddRadiusIntersectFilter](#)

[ClearFilters](#)

[IsPointWithinFilters](#)

[IsZipCodeWithinFilters](#)

## AddRadiusExcludeFilter

Syntax

```
AddRadiusExcludeFilter(bstrZipCode, bstrDistance)  
long AddRadiusExcludeFilter(BSTR bstrZipCode, BSTR bstrDistance)
```

Description

Add a filter that excludes all points within a specified radius of a ZIP Code.

Parameters

*bstrZipCode as String*

The ZIP Code whose latitude and longitude coordinates define the origin of the exclusion radius.

*bstrDistance as String*

The radius of the circle, in miles.

*bstrZipCode*

The ZIP Code whose latitude and longitude coordinates define the origin of the exclusion radius.

*bstrDistance*

The radius of the circle, in miles.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This filter will exclude all points, as defined by their latitude and longitude coordinates, that lie within a circle that is centered on the center point of the specified ZIP Code.

To use this filter, call either `IsPointWithinFilters` or `IsZipCodeWithinFilters`. Those functions will check to see if a point or ZIP Code is within the areas defined by the currently applied filters.

The ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will result in more precise coordinates.

## See also

[AddIsBetweenFilter](#)

[AddRadiusIncludeFilter](#)

[AddRadiusIntersectFilter](#)

[ClearFilters](#)

[IsPointWithinFilters](#)

[IsZipCodeWithinFilters](#)

## AddRadiusIncludeFilter

### Syntax

```
AddRadiusIncludeFilter (bstrZipCode, bstrDistance)
```

```
long AddRadiusIncludeFilter(BSTR bstrZipCode, BSTR bstrDistance)
```

### Description

Add a filter that includes all points within a radius from the origin of a ZIP Code.

### Parameters

*bstrZipCode as String*

The ZIP Code whose latitude and longitude coordinates define the origin of the inclusion radius.

*bstrDistance as String*

The radius of the circle, in miles.

*bstrZipCode*

The ZIP Code whose latitude and longitude coordinates define the origin of the inclusion radius.

*bstrDistance*

The radius of the circle, in miles.

## Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This filter will exclude all points, as defined by their latitude and longitude coordinates, that lie within a circle that is centered on the center point of the specified ZIP Code.

To use this filter, call either `IsPointWithinFilters` or `IsZipCodeWithinFilters`. Those functions will check to see if a point or ZIP Code is within the areas defined by the currently applied filters.

The ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will result in more precise coordinates.

## See also

[AddIsBetweenFilter](#)

[AddRadiusExcludeFilter](#)

[AddRadiusIntersectFilter](#)

[ClearFilters](#)

[IsPointWithinFilters](#)

[IsZipCodeWithinFilters](#)

## AddRadiusIntersectFilter

## Syntax

```
AddRadiusIntersectFilter(bstrZipCode, bstrDistance)
```

```
long AddRadiusIntersectFilter(BSTR bstrZipCode, BSTR bstrDistance)
```

#### Description

Adds a filter that includes all points that have been selected by previous filters and are within a radius from the midpoint of a ZIP Code.

#### Parameters

*bstrZipCode as String*

The ZIP Code whose latitude and longitude coordinates define the origin of the inclusion radius.

*bstrDistance as String*

The radius of the circle, in miles.

*bstrZipCode*

The ZIP Code whose latitude and longitude coordinates define the origin of the inclusion radius.

*bstrDistance*

The radius of the circle, in miles.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This filter will include all points selected by previous filters that also are within the specified radius from the origin ZIP Code. For example, calling `AddRadiusIncludeFilter` followed by `AddRadiusIntersectFilter` will return addresses that are common to both circles.

To use this filter, call either `IsPointWithinFilters` or `IsZipCodeWithinFilters`. Those functions will check to see if a point or ZIP Code is within the areas defined by the currently applied filters.

The origin ZIP Code can be anywhere from 3 to 9 digits.

#### See also

[AddIsBetweenFilter](#)

[AddRadiusExcludeFilter](#)

[AddRadiusIncludeFilter](#)

[ClearFilters](#)

[IsPointWithinFilters](#)

[IsZipCodeWithinFilters](#)

## BearingDegreesFromBaseToPoint

### Syntax

```
BearingDegreesFromBaseToPoint(bstrLatitude, bstrLongitude)
```

```
BSTR BearingDegreesFromBaseToPoint(BSTR bstrLatitude, BSTR bstrLongitude, BSTR *pbstrBearingAsDegree)
```

### Description

Returns the angle of the input coordinates relative to a base point.

### Parameters

- `bstrLatitude` as String – The latitude coordinate of the point to compare to the base point.
- `bstrLongitude` as String – The longitude coordinate of the point to compare to the base point.
- `bstrLatitude` – The latitude coordinate of the point to compare to the base point.
- `bstrLongitude` – The longitude coordinate of the point to compare to the base point.
- `pbstrBearingAsDegree` – Returns the angle of the point relative to the base point, in degrees; -1 if unable to determine

### Return values

The angle of the point relative to the base point, in degrees; -1 if unable to determine (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

The angle is measured counterclockwise from a reference line running east-west to a line drawn through the point defined by the input coordinates and the base point.

The base point is defined by the coordinates of the midpoint of the ZIP Code contained in the ZipCode property.

## See also

[BearingDegreesFromBaseToZipCode](#)

[ZipCode](#)

## BearingDegreesFromBaseToZipCode

### Syntax

```
BearingDegreesFromBaseToZipCode (bstrZipCode)
```

```
BSTR BearingDegreesFromBaseToZipCode (BSTR bstrZipCode, BSTR *pbstrBearingAsDegree)
```

### Description

Returns the angle of the midpoint of the input ZIP Code relative to a base point.

### Parameters

*bstrZipCode as String*

The ZIP Code whose midpoint coordinates you want to compare to the base point.

*bstrZipCode*

The ZIP Code whose midpoint coordinates you want to compare to the base point.

*pbstrBearingAsDegree*

Returns the angle of the point relative to the base point, in degrees; -1 if unable to determine.

### Return values

The angle of the point relative to the base point, in degrees; -1 if unable to determine (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

The angle is measured counterclockwise from a reference line running east-west to a line drawn through the point defined by the coordinates that represent the input ZIP code and the base point.

The base point is defined by the coordinates of the midpoint of the ZIP Code contained in the `ZipCode` property.

The input ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will result in more precise coordinates.

## See also

[BearingDegreesFromBaseToPoint](#)

[ZipCode](#)

## BearingDirectionFromBaseToPoint

### Syntax

```
BearingDirectionFromBaseToPoint(bstrLatitude, bstrLongitude)
```

```
BSTR BearingDirectionFromBaseToPoint(BSTR bstrLatitude, BSTR bstrLongitude, BSTR *pbstrBearingAsDirection)
```

### Description

Direction of the input coordinates relative to a base point.

### Parameters

*bstrLatitude as String*

The latitude coordinate of the point to compare to the base point.

*bstrLongitude as String*

The longitude coordinate of the point to compare to the base point.

*bstrLatitude*

The latitude coordinate of the point to compare to the base point.

*bstrLongitude*

The longitude coordinate of the point to compare to the base point.

*pbstrBearingAsDirection*

Returns the direction relative to the base point; -1 if unable to determine.

#### Return values

The direction of the point relative to the base point; -1 if unable to determine (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

The direction corresponds to an angle that is measured counterclockwise from a reference line running east-west to a line drawn through the point defined by the input coordinates and the base point.

The base point is defined by the coordinates of the midpoint of the ZIP Code contained in the ZipCode property.

Possible values are listed in the following table:

Bearing Angle	Bearing Direction
Between 348.75° and 11.25°, inclusive	E
Between 11.25° and 33.75°	ENE
Between 33.75° and 56.25°, inclusive	NE
Between 56.25° and 78.75°	NNE
Between 78.75° and 101.25°, inclusive	N
Between 101.25° and 123.75°	NNW
Between 123.75° and 146.25°, inclusive	NW
Between 146.25° and 168.75°	WNW
Between 168.75° and 191.25°, inclusive	W
Between 191.25° and 213.75°	WSW
Between 213.75° and 236.25°, inclusive	SW
Between 236.25° and 258.75°	SSW

Bearing Angle	Bearing Direction
Between 258.75° and 281.25°, inclusive	S
Between 281.25° and 303.75°	SSE
Between 303.75° and 326.25°, inclusive	SE
Between 326.25° and 348.75°	ESE

See also

[BearingDirectionFromBaseToZipCode](#)

[ZipCode](#)

## BearingDirectionFromBaseToZipCode

Syntax

```
BearingDirectionFromBaseToZipCode (bstrZipCode)
```

```
BSTR BearingDirectionFromBaseToZipCode (BSTR bstrZipCode, BSTR *pbstrBearingAsDirection)
```

Description

Returns the direction of the midpoint of a ZIP Code relative to a base point.

Parameters

*bstrZipCode as String*

The ZIP Code to compare to the base point.

*bstrZipCode*

The ZIP Code to compare to the base point.

*pbstrBearingAsDirection*

Returns the direction relative to the base point; -1 if indeterminate.

Return values

The direction of the point relative to the base point; -1 if unable to determine (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

The direction corresponds to an angle that is measured counterclockwise from a reference line running east-west to a line drawn through the point defined by the coordinates that represent the input ZIP code and the base point.

The base point is defined by the coordinates of the midpoint of the ZIP Code contained in the ZipCode property.

The input ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will result in more precise coordinates.

Possible values are listed in the following table:

Bearing Angle	Bearing Direction
Between 348.75° and 11.25°, inclusive	E
Between 11.25° and 33.75°	ENE
Between 33.75° and 56.25°, inclusive	NE
Between 56.25° and 78.75°	NNE
Between 78.75° and 101.25°, inclusive	N
Between 101.25° and 123.75°	NNW
Between 123.75° and 146.25°, inclusive	NW
Between 146.25° and 168.75°	WNW
Between 168.75° and 191.25°, inclusive	W
Between 191.25° and 213.75°	WSW
Between 213.75° and 236.25°, inclusive	SW
Between 236.25° and 258.75°	SSW
Between 258.75° and 281.25°, inclusive	S

Bearing Angle	Bearing Direction
Between 281.25° and 303.75°	SSE
Between 303.75° and 326.25°, inclusive	SE
Between 326.25° and 348.75°	ESE

See also

[BearingDirectionFromBaseToPoint](#)

[ZipCode](#)

## CheckGeocode

Syntax

```
CheckGeocode
```

```
long CheckGeocode()
```

Description

Retrieves geocode information for the ZIP Code contained in the ZipCode property.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

This function will populate the various geocode properties for the ZIP Code contained in the ZipCode property.

See also

[CensusBlock](#)

[CensusTract](#)

[CountyFIPSCode](#)

[Footnotes](#)

[Latitude](#)

[Longitude](#)

[MSACode](#)

[RecordType](#)

## ClearFilters

Syntax

```
ClearFilters  
long ClearFilters()
```

Description

Removes all currently applied filters.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [\\_COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [\\_COM MRTKTaskLib result code](#).

Notes

This function will remove all applied filters.

See also

[AddIsBetweenFilter](#)

[AddRadiusExcludeFilter](#)

[AddRadiusIncludeFilter](#)

[AddRadiusIntersectFilter](#)

## ClearGeocode

Syntax

```
ClearGeocode  
long ClearGeocode()
```

Description

Clears all properties.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.  
0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

This function will reset all properties to empty strings.

See also

[CensusBlock](#)

[CensusTract](#)

[CountyFIPSCode](#)

[Footnotes](#)

[Latitude](#)

[Longitude](#)

[MSACode](#)

[RecordType](#)

[ZipCode](#)

## ConvertDoubleToString

### Syntax

```
ConvertDoubleToString(dDouble)  
long ConvertDoubleToString(double dDouble, BSTR *pbstrForDouble)
```

### Description

Converts a double data type to a string.

### Parameters

*dDouble as Double*

The value to convert.

*dDouble*

The value to convert.

*pbstrForDouble*

Returns the converted value as a string.

### Return values

The converted value (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

### See also

[ConvertStringToDouble](#)

## ConvertLongToString

### Syntax

```
ConvertLongToString(lLong)  
long ConvertLongToString(long lLong, BSTR *pbstrForLong)
```

### Description

Converts a long data type to a string.

### Parameters

*lLong as Long*

The value to convert.

*lLong*

The value to convert.

*pbstrForLong*

Returns the converted value as a string.

### Return values

The converted value (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### See also

[ConvertStringToLong](#)

## ConvertMilesToKilometers

### Syntax

```
ConvertMilesToKilometers(bstrMiles)  
long ConvertMilesToKilometers(BSTR bstrMiles, BSTR *pbstrKilometers)
```

### Description

Converts miles to kilometers.

### Parameters

*bstrMiles as String*

The value to convert.

*bstrMiles*

The value to convert.

*pbstrKilometers*

Returns the converted value in kilometers.

#### Return values

The converted value (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## ConvertStringToDouble

#### Syntax

```
ConvertStringToDouble (bstrString)
```

```
long ConvertStringToDouble (BSTR bstrString, double *pDouble)
```

#### Description

Converts a string data type to a double.

#### Parameters

*bstrString as String*

The value to convert.

*bstrString –*

The value to convert.

*pDouble*

Returns the converted value.

#### Return values

The converted value (as Double). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

See also

[ConvertDoubleToString](#)

## ConvertStringToLong

Syntax

```
ConvertStringToLong(bstrString)
long ConvertStringToLong(BSTR bstrString, long *pLong)
```

Description

Converts a string data type to a long.

Parameters

*bstrString as String*

The value to convert.

*bstrString*

The value to convert.

*pLong*

Returns the converted value.

Return values

The converted value (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

See also

[ConvertLongToString](#)

## DistanceFromBaseToPoint

Syntax

```
DistanceFromBaseToPoint(bstrLatitude, bstrLongitude)
BSTR DistanceFromBaseToPoint(BSTR bstrLatitude, BSTR bstrLongitude,
BSTR *pbstrDistance)
```

## Description

Returns the distance from a coordinate pair to the ZIP Code specified in the ZipCode property.

## Parameters

### *bstrLatitude as String*

The latitude coordinate of the point to compare.

### *bstrLongitude as String*

The longitude coordinate of the point to compare.

### *bstrLatitude*

The latitude coordinate of the point to compare.

### *bstrLongitude*

The longitude coordinate of the point to compare.

### *pbstrDistance*

Returns the distance from the coordinate pair to the base point, in miles; -1 if unable to determine.

## Return values

The distance of the point from the base point, in miles; -1 if unable to determine (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

The base point is defined by the coordinates that represent the midpoint of the ZIP Code contained in the ZipCode property.

## See also

[DistanceFromBaseToZipCode](#)

[ZipCode](#)

## DistanceFromBaseToZipCode

## Syntax

```
DistanceFromBaseToZipCode (bstrZipCode)
```

```
BSTR DistanceFromBaseToZipCode (BSTR bstrZipCode, BSTR *pb-  
strDistance)
```

#### Description

Distance from the midpoint of the input ZIP Code to the ZIP Code specified in the ZipCode property.

#### Parameters

*bstrZipCode as String*

The ZIP Code whose coordinates represent the point to compare.

*bstrZipCode*

The ZIP Code whose coordinates represent the point to compare.

*pbstrDistance*

Returns the distance from the input point to the base point in miles; -1 if unable to determine.

#### Return values

The distance from the input point to the base point in miles; -1 if unable to determine (as String). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

The base point is defined by the coordinates that represent the ZIP Code contained in the ZipCode property.

The input ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will create more accurate results.

#### See also

[DistanceFromBaseToPoint](#)

[ZipCode](#)

#### IsPointWithinFilters

#### Syntax

```
IsPointWithinFilters (bstrLatitude, bstrLongitude)
```

```
long IsPointWithinFilters(BSTR bstrLatitude, BSTR bstrLongitude,  
    BOOL *pbInFilters)
```

#### Description

Indicates if the input coordinate pair is within the area described by the currently applied filters.

#### Parameters

##### *bstrLatitude as String*

The latitude coordinate of the point to analyze.

##### *bstrLongitude as String*

The longitude coordinate of the point to analyze.

##### *bstrLatitude*

The latitude coordinate of the point to analyze.

##### *bstrLongitude*

The longitude coordinate of the point to analyze.

##### *pbInFilters*

Returns 0 if False, 1 if True; -1 if unable to determine.

#### Return values

0 if False, 1 if True; -1 if unable to determine (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This function indicates if the input point lies within the area included by the applied filters.

#### See also

[AddIsBetweenFilter](#)

[AddRadiusExcludeFilter](#)

[AddRadiusIncludeFilter](#)

[AddRadiusIntersectFilter](#)

[ClearFilters](#)

[IsZipCodeWithinFilters](#)

## IsZipCodeWithinFilters

### Syntax

```
IsZipCodeWithinFilters (bstrZipCode)  
long IsZipCodeWithinFilters (BSTR bstrZipCode, BOOL *pbInFilters)
```

### Description

Indicates if the input ZIP Code is within the area described by the currently applied filters.

### Parameters

*bstrZipCode as String*

The ZIP Code of the point to analyze.

*bstrZipCode*

The ZIP Code of the point to analyze.

*pbInFilters*

Returns 0 if False, 1 if True; -1 if unable to determine.

### Return values

0 if False, 1 if True; -1 if unable to determine (as Long). The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

This function indicates if the point defined by the coordinates that represent the input ZIP Code lies within the area selected by the applied filters.

The input ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will create more accurate results.

See also

[AddIsBetweenFilter](#)

[AddRadiusExcludeFilter](#)

[AddRadiusIncludeFilter](#)

[AddRadiusIntersectFilter](#)

[ClearFilters](#)

[IsPointWithinFilters](#)

## EndTask

Syntax

```
EndTask
```

```
long EndTask()
```

Description

When finished with the task, clean up and release and resources used.

Parameters

None.

Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

Call EndTask when you are done with the GeocodeTask object.

## GeocodeTask Properties

The GeocodeTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

## CountyFIPSCode

### Syntax

```
CountyFIPSCode  
long CountyFIPSCode (BSTR *pVal)
```

### Description

The county FIPS code associated with a ZIP Code. Output only.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

This property contains the full 5-digit County FIPS code, where the first two digits represent the state and the last three digits represent the county.

Data returned is for the ZIP Code contained in the ZipCode property.

### See also

[CheckGeocode](#)

[ZipCode](#)

## Footnotes

### Syntax

```
Footnotes
long Footnotes(BSTR *pVal)
```

### Description

Contains additional information about a geocode lookup.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

Data returned is for the ZIP Code contained in the ZipCode property.

This property returns a code that indicates the granularity of ZIP Code match for a geocode lookup, as described in the following table:

Value	Description
00	Unable to perform a Geocode lookup.
03	Geocode data based on a 3-digit ZIP Code.
05	Geocode data based on a 5-digit ZIP Code.

Value	Description
07	Geocode data based on a 7-digit ZIP Code.
09	Geocode data based on a 9-digit ZIP Code.

See also

[CheckGeocode](#)

[ZipCode](#)

## Latitude

### Syntax

```
Latitude  
long Latitude(BSTR *pVal)
```

### Description

Contains the latitude coordinate of a ZIP Code.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

Data returned is for the ZIP Code contained in the ZipCode property.

See also

[CheckGeocode](#)

[ZipCode](#)

## Longitude

### Syntax

```
Longitude  
long Longitude(BSTR *pVal)
```

### Description

Retrieves the longitude coordinate of a ZIP Code.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

Data returned is for the ZIP Code contained in the ZipCode property.

See also

[CheckGeocode](#)

[ZipCode](#)

## MSACode

Syntax

MSACode

```
long MSACode (BSTR *pVal)
```

Description

Retrieves the Metropolitan Statistical Area code of a ZIP Code.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

Data returned is for the ZIP Code contained in the ZipCode property.

See also

[CheckGeocode](#)

[ZipCode](#)

## RecordType

Syntax

```
RecordType
```

```
long RecordType (BSTR *pVal)
```

Description

Retrieves the record type of a ZIP Code.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

This property returns a USPS flag that represents the address type for the ZIP Code that is contained in the ZipCode property.

For example, a PO Box address will return a value of "P."

The following record types are currently valid:

- S – Street record
- P – Post Office box
- R – Rural Route or Highway Contract
- H – High-rise, Building or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

See also

[CheckGeocode](#)

[ZipCode](#)

## CensusBlock

Syntax

```
CensusBlock
```

```
long CensusBlock(BSTR *pVal)
```

Description

Retrieves the census block data for a ZIP Code.

Data Type

String

BSTR

Parameters

None

pVal – Returns property value.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

Data returned is for the ZIP Code contained in the ZipCode property.

### See also

[CheckGeocode](#)

[CensusTract](#)

[ZipCode](#)

## CensusTract

### Syntax

```
CensusTract  
long CensusTract (BSTR *pVal)
```

### Description

Retrieves the census tract data for a ZIP Code.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can only read this property.

Call CheckGeocode before attempting to retrieve the value of this property.

Data returned is for the ZIP Code contained in the ZipCode property.

## See also

[CheckGeocode](#)

[CensusBlock](#)

[ZipCode](#)

## ZipCode

### Syntax

```
ZipCode  
long ZipCode(BSTR *pVal) /* retrieving */  
long ZipCode(BSTR newVal) /* setting */
```

### Description

Contains the ZIP Code.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

The CheckGeocode function will provide information based on the value of this property.

This property defines the base point used by the bearing and distance functions.

The input ZIP Code can be anywhere from 3 to 9 digits. More complete ZIP Codes will give in more accurate results.

## See also

[BearingDegreesFromBaseToPoint](#)

[BearingDegreesFromBaseToZipCode](#)

[BearingDirectionFromBaseToPoint](#)

[BearingDirectionFromBaseToZipCode](#)

[CheckGeocode](#)

[DistanceFromBaseToPoint](#)

[DistanceFromBaseToZipCode](#)

[CensusBlock](#)

[CensusTract](#)

[CountyFIPSCode](#)

[Footnotes](#)

[Latitude](#)

[Longitude](#)

[MSACode](#)

[RecordType](#)

## The COM PhoneTask Object for Parsing or Building Phone Numbers

The BCC Architect PhoneTask object can parse a phone number into its individual elements. It can also build a phone number from its individual elements, using several different formatting options.

### PhoneTask Functions

The PhoneTask functions are defined below. Once you have added a reference to the BCC Architect files, you can begin using these functions in your project.

#### PrepareTask

##### Syntax

```
PrepareTask  
  
long PrepareTask()
```

##### Description

Initialize and prepare the object.

##### Parameters

None.

##### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

##### Notes

PrepareTask should be called only once, when the PhoneTask object is first created.

This function should be called before you call any of the other functions or set any properties of PhoneTask.

#### ClearPhone

##### Syntax

```
ClearPhone
```

```
long ClearPhone()
```

#### Description

Reset the phone number elements to null.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

This function will quickly clear the phone number information from a PhoneTask object.

This function can be used to bring the PhoneTask object into a known state (i.e., no area code, prefix, etc.) before setting the object's properties.

#### See also

[Parse Phone](#)

[Format Phone](#)

## FormatPhone

#### Syntax

```
FormatPhone
```

```
long FormatPhone()
```

#### Description

Builds and formats a phone number from its individual elements.

#### Parameters

None.

#### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function builds a phone number from its individual elements and updates the value of the Number property.

The number is formatted based on the value of the Format property.

If an error is encountered during the build process, the ErrorCode property will be updated to reflect the problem.

## See also

[AreaCode](#)

[Prefix](#)

[LineNumber](#)

[Extension](#)

[Number](#)

[Format](#)

[ErrorCode](#)

## ParsePhone

### Syntax

```
ParsePhone  
long ParsePhone()
```

### Description

Parses a phone number into its individual elements.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

This function parses the value of the Number property into its individual elements.

If an error is encountered during the process, the ErrorCode property will be updated to reflect the problem.

## See also

[AreaCode](#)

[Prefix](#)

[LineNumber](#)

[Extension](#)

[ErrorCode](#)

## EndTask

### Syntax

```
EndTask  
  
long EndTask()
```

### Description

When finished with the task, clean up and release and resources used.

### Parameters

None.

### Return values

Nothing. The VB Err object will contain the [\\_COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [\\_COM MRTKTaskLib result code](#).

## Notes

EndTask releases the PhoneTask object.

We recommend that you call EndTask when you are done with the PhoneTask object.

You do not need to call EndTask after each call to ParsePhone or FormatPhone.

## PhoneTask Properties

The PhoneTask properties are defined below. Once you have added a reference to the BCC Architect files, you can begin using these properties in your project.

### AreaCode

#### Syntax

```
AreaCode  
  
long AreaCode(BSTR *pVal) /* retrieving */  
long AreaCode(BSTR newVal) /* setting */
```

#### Description

Sets and retrieves the area code of a phone number.

#### Data Type

String

BSTR

#### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

#### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

The FormatPhone function will create a phone number in the Number property using this property, along with the Prefix, LineNumber and Extension properties.

The ParsePhone function will populate this property from a properly formatted phone number in the Number property.

See also

[ParsePhone](#)

[Format Phone](#)

## ErrorCode

Syntax

```
ErrorCode  
long ErrorCode(long *pVal)
```

Description

Retrieves the error code associated with the last call to either ParsePhone or FormatPhone.

Data Type

Long

long

Parameters

None

pVal – Returns property value.

Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

Notes

You can only read this property.

The value of this property is updated when you call ParsePhone or FormatPhone.

A zero value indicates the function did not encounter any problems.

An error is indicated according to the following table:

Error ID	Enum Value
phPHONE_ERROR_NUMBER_BAD_LENGTH	-1
phPHONE_ERROR_NUMBER_INTERNATIONAL	-2
phPHONE_ERROR_NUMBER_EMPTY	-3
phPHONE_ERROR_BADFORMAT	-5

See also

[Parse Phone](#)

[Format Phone](#)

## Extension

### Syntax

```
Extension
long Extension(BSTR *pVal) /* retrieving */
long Extension(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the extension of a phone number.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

#### Notes

You can read and set this property.

The FormatPhone function will create a phone number in the Number property using this property, along with the AreaCode, Prefix and LineNumber properties.

The ParsePhone function will populate this property from a properly formatted phone number in the Number property.

#### See also

[Parse Phone](#)

[Format Phone](#)

#### Format

##### Syntax

```
Format  
  
long Format(long *pVal) /* retrieving */  
long Format(long newVal) /* setting */
```

##### Description

Determines the format of the phone number.

##### Data Type

Long

long

##### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

## Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

## Notes

You can read and set this property.

The value of this property determines the format of the Number property created by FormatPhone and expected by ParsePhone.

Possible values are shown in the following table:

Format ID	Format
phPHONE_FORMAT_1	XXXYYYZZZZ
phPHONE_FORMAT_2	(XXX) YYY-ZZZZ
phPHONE_FORMAT_3	XXX-YYY-ZZZZ
phPHONE_FORMAT_4	XXX.YYY.ZZZZ
phPHONE_FORMAT_5	XXX/YYY-ZZZZ
phPHONE_FORMAT_6	YYY-ZZZZ
phPHONE_FORMAT_7	1(XXX) YYY-ZZZZ
phPHONE_FORMAT_8	1-XXX-YYY-ZZZZ
phPHONE_FORMAT_9	1.XXX.YYY.ZZZZ
phPHONE_FORMAT_10	1XXX/YYY-ZZZZ

## See also

[Format Phone](#)

[Number](#)

## LineNumber

### Syntax

```
LineNumber  
  
long LineNumber(BSTR *pVal) /* retrieving */  
long LineNumber(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the line number portion of a phone number.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

The line number is the last four digits of the phone number.

The FormatPhone function will create a phone number in the Number property using this property, along with the AreaCode, Prefix and Extension properties.

The ParsePhone function will populate this property from a properly formatted phone number in the Number property.

See also

[Parse Phone](#)

[Format Phone](#)

## Number

### Syntax

Number

```
long Number(BSTR *pVal) /* retrieving */
```

```
long Number(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the phone number.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

This property is parsed into its individual elements when you call ParsePhone.

The FormatPhone function will combine the AreaCode, Prefix, LineNumber and Extension properties into a phone number in this property. It will be formatted according to the value of the Format property.

See also

[Parse Phone](#)

[Format Phone](#)

## Prefix

### Syntax

```
Prefix  
  
long Prefix(BSTR *pVal) /* retrieving */  
long Prefix(BSTR newVal) /* setting */
```

### Description

Sets and retrieves the prefix of a phone number.

### Data Type

String

BSTR

### Parameters

None

pVal – Returns property value.

newVal – Value to assign to the property.

### Return values

Property value. The VB Err object will contain the [COM MRTKTaskLib result code](#) if an error occurs.

0 if successful, otherwise a [COM MRTKTaskLib result code](#).

### Notes

You can read and set this property.

The prefix is the first three digits of the phone number, not including the area code.

The FormatPhone function will create a phone number in the Number property using this property, along with the AreaCode, LineNumber and Extension properties.

The ParsePhone function will populate this property from a properly formatted phone number in the Number property.

See also

[Parse Phone](#)

[Format Phone](#)

## COM Field Names

Below are all the field names for an input or output record. These are concatenated together with a delimiter and passed to the SetProperty function for either the mrtkFIELD\_LIST\_IN or mrtkFIELD\_LIST\_OUT properties. These Field IDs specify which fields will be passed in or returned from a particular task.

Field ID	Data Direction	Data Type Returned	Description
FLD_ADDRESS_BLOCK	In/Out	String	A block of addresses; see AddressBlock property of ZipTask for more information.

Field ID	Data Direction	Data Type Returned	Description									
FLD_ADDRESS_SUGGESTION_LIST	Out	String	<p>A list of suggested addresses that is returned based on an input address that is invalid and can't be corrected. Data is output to the field only if it is included in the output field list for batch address correction.</p> <p>The field will display a maximum of 10 entries, all of which have had their deliverability confirmed (DPV).</p> <p>By default, the following delimiters are used in the output text:</p> <table border="1"> <thead> <tr> <th>Delimiter Type</th> <th>ASCII</th> <th>Hex</th> </tr> </thead> <tbody> <tr> <td>Field</td> <td>25</td> <td>19</td> </tr> <tr> <td>Record</td> <td>26</td> <td>1A</td> </tr> </tbody> </table> <p><b>IMPORTANT</b> You may have already set these delimiters to be used globally via the global properties <code>mrtkDELIMITER_FIELD</code> and <code>mrtkDELIMITER_RECORD</code>, respectively. If this is the case, when the <code>ValidateProperties</code> function is run, an error is generated.</p> <p>As a solution, there are two additional properties that you can use to set delimiters:  <code>mrtkSECONDARY_DELIMITER_FIELD</code> and <code>mrtkSECONDARY_DELIMITER_RECORD</code>.</p> <p>The type of input addresses that will generally return suggestions are those that have certain address elements that are missing or invalid, such as street suffixes and street directionals. Suggestions return the following fields:</p> <ul style="list-style-type: none"> <li>• AddressLine 1</li> <li>• City</li> </ul>	Delimiter Type	ASCII	Hex	Field	25	19	Record	26	1A
Delimiter Type	ASCII	Hex										
Field	25	19										
Record	26	1A										

Field ID	Data Direction	Data Type Returned	Description
FLD_ADDRESSLINE1	In/Out	String	The primary address line; required for CASS processing unless FLD_ADDRESSLINE2 is specified.
FLD_ADDRESSLINE2	In/Out	String	The secondary address line; required for CASS processing unless FLD_ADDRESSLINE1 is specified.
FLD_AFTER_CASS_ADDRESSLINE2	Out	String	Address Line 2 of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_ADDRESSLINE1	Out	String	Address Line 1 of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_CARRIER_ROUTE	Out	String	Carrier route for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_CITY	Out	String	City of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_CITY_ABBREVIATED	Out	String	Abbreviated city of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_COUNTY_CODE	Out	String	County code for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_COUNTY_NAME	Out	String	County name for the input address after address correction and before NCOA processing.

Field ID	Data Direction	Data Type Returned	Description
FLD_AFTERZ4_DPBARCODE	Out	String	Delivery Point barcode string for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_DPC	Out	String	DPC for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_DPV_CODED	Out	String	After address correction and before NCOA processing, indicates if the address was confirmed with DPV.
FLD_AFTERZ4_DPV_DOOR_NOT_ACCESSIBLE	Out	Boolean	After address correction and before NCOA processing, returns a DPV code that indicates that the door is not physically accessible by mail carrier.

Field ID	Data Direction	Data Type Returned	Description
FLD_AFTERZ4_ DPV_FOOTNOTE	Out	String	<p>After address correction and before NCOA processing, the DPV footnote code:</p> <ul style="list-style-type: none"> <li>LK Processing locked out due to a seed record being processed.</li> <li>AA Matched to the ZIP+4 file</li> <li>A1 No match against the ZIP+4 file</li> <li>BB Matched to DPV file (all components confirmed)</li> <li>CC Matched only after removing secondary information; they were presented but invalid.</li> <li>N1 Input primary matched, but high-rise missing secondary number.</li> <li>M1 Primary number missing.</li> <li>M3 Primary number invalid.</li> <li>P1 Input missing PO, RR or HC box number.</li> <li>P3 Failed DPV because of invalid PO, RR or HC box number.</li> <li>RR Matched CMRA (found in CMRA file).</li> <li>R1 Matched CMRA, but secondary number (i.e., PMB) missing.</li> <li>U1 Matched unique ZIP Code.</li> <li>G1 Matched general delivery.</li> <li>F1 Matched military address.</li> </ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_AFTERZ4_DPV_INDICATOR	Out	String	<p>After address correction and before NCOA processing, returns a single character that indicates the level of address validity with DPV:</p> <ul style="list-style-type: none"><li>• Y - Both the primary and secondary (if present) validated against the DPV database.</li><li>• S - The primary address is valid according to DPV, but the secondary is invalid.</li><li>• D - The primary address is valid according to DPV, but the address is missing secondary information.</li><li>• N - The primary address is not valid according to DPV.</li><li>• "" - The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.</li><li>• X - The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.</li><li>• E - The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.</li></ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_AFTERZ4_DPV_IS_CMRA	Out	String	Indicates if the address after correction and before NCOA processing is a commercial mail-receiving agent.
FLD_AFTERZ4_DPV_IS_NOSTAT	Out	String	Indicates if the address, after address correction and before NCOA processing, is not receiving delivery and is not counted as a possible delivery. The address is not receiving delivery because: 1) delivery has not been established, 2) The customer receives mail as part of a drop, 3) the carrier destroys or returns all of the mail.
FLD_AFTERZ4_DPV_IS_PBSA	Out	String	Indicates if address, after address correction and before NCOA processing, is a Post Office Box street address (PBSA).
FLD_AFTERZ4_DPV_IS_THROWBACK	Out	Boolean	After address correction and before NCOA processing, returns a DPV code that indicates that delivery is made to a PO Box address.
FLD_AFTERZ4_DPV_IS_VACANT	Out	String	Indicates if address, after address correction and before NCOA processing, is unoccupied.
FLD_AFTERZ4_DPV_NO_SECURE_LOCATION	Out	Boolean	After address correction and before NCOA processing, returns a DPV code that indicates that the location is not secure.
FLD_AFTERZ4_ERROR_CODE	Out	String	Error code for the input address after address correction and before NCOA processing.

Field ID	Data Direction	Data Type Returned	Description
FLD_AFTERZ4_LOT_NUMBER	Out	String	LOT number for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_POST_DIRECTIONAL	Out	String	Post-directional of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_PO_BOX_DELIVERY_ONLY_ZIP	Out	String	Indicates whether delivery is to an address in a PO Box only zone, after address correction and before NCOA processing.
FLD_AFTERZ4_PRE_DIRECTIONAL	Out	String	Predirectional of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_PRIMARY_NUMBER	Out	String	Primary number of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_RECORD_TYPE	Out	String	Record type for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_STATE	Out	String	State for the input address after address correction and before NCOA processing.
FLD_AFTERZ4_STREET_NAME	Out	String	Street name of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_SUFFIX	Out	String	Suffix of the input address after address correction and before NCOA processing.

Field ID	Data Direction	Data Type Returned	Description
FLD_AFTERZ4_UNIT_DESIGNATOR	Out	String	Unit designator of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_UNIT_NUMBER	Out	String	Unit number of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_URBANIZATION	Out	String	Urbanization of the input address after address correction and before NCOA processing.
FLD_AFTERZ4_ZIPCODE	Out	String	ZIP Code of the input address after address correction and before NCOA processing.
FLD_BREAK_MARK	Out	String	Returns values for ptBREAK_MARK_IND_BUNDLE, ptBREAK_MARK_IND_CONTAINER, and ptBREAK_MARK_IND_PALLET for records that mark a bundle break, container break, or pallet break. The break marks returned are set by ptAPPLY_BREAK_MARK_INDICATOR
FLD_BUSINESS	In/Out	String	Business name.
FLD_CARRIER_ROUTE	In/Out	String	Carrier route number.
FLD_CASSDATE	In/Out	Long	Contains information about when address was last processed.
FLD_CITY	In/Out	String	City; required for CASS processing unless FLD_LASTLINE specified.
FLD_CITY_ABBREVIATED	Out	String	Returns the abbreviated city name.

Field ID	Data Direction	Data Type Returned	Description
FLD_CONGRESSIONAL_DISTRICT	Out	String	Congressional district number.
FLD_COUNTRY	In/Out	String	Country.
FLD_COUNTY_CODE	Out	Long	5-digit county code.
FLD_COUNTY_NAME	Out	String	County name.
FLD_DELETE_RECORD_FLAG	Out	BOOL	Not currently implemented.
FLD_DP_BARCODE	In/Out	String	12-digit number representing the either the IM barcode or the Delivery Point Barcode, preceded and followed by a colon. If you set pwPRESORT_INTELLIGENT_MAIL_ONLY to TRUE, CASS Task and ZIP Task will not update this field.
FLD_DPC	In/Out	String	Delivery Point/Check Digit.
FLD_DPV_CODED	Out	Long	Indicates if the address was confirmed with DPV.
FLD_DPV_DOOR_NOT_ACCESSIBLE	Out	Boolean	Indicates addresses where carriers cannot knock on the door to deliver mail that will not fit into a mailbox or where carriers cannot physically access a residence/building.

Field ID	Data Direction	Data Type Returned	Description
FLD_DPV_ FOOTNOTE	Out	String	<p>DPV footnote code:</p> <ul style="list-style-type: none"> <li>LK Processing locked out due to a seed record being processed.</li> <li>AA Matched to the ZIP+4 file</li> <li>A1 No match against the ZIP+4 file</li> <li>BB Matched to DPV file (all components confirmed)</li> <li>CC Matched only after removing secondary information; they were presented but invalid.</li> <li>N1 Input primary matched, but high-rise missing secondary number.</li> <li>M1 Primary number missing.</li> <li>M3 Primary number invalid.</li> <li>P1 Input missing PO, RR or HC box number.</li> <li>P3 Failed DPV because of invalid PO, RR or HC box number.</li> <li>RR Matched CMRA (found in CMRA file).</li> <li>R1 Matched CMRA, but secondary number (i.e., PMB) missing.</li> <li>U1 Matched unique ZIP Code.</li> <li>G1 Matched general delivery.</li> <li>F1 Matched military address.</li> </ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_DPV_INDICATOR	Out	String	<p>Returns a single character that indicates the level of address validity with DPV:</p> <ul style="list-style-type: none"><li>• Y - Both the primary and secondary (if present) validated against the DPV database.</li><li>• S - The primary address is valid according to DPV, but the secondary is invalid.</li><li>• D - The primary address is valid according to DPV, but the address is missing secondary information.</li><li>• N - The primary address is not valid according to DPV.</li><li>• "" - The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.</li><li>• X - The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.</li><li>• E - The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.</li></ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_DPV_IS_CMRA	Out	Long	Indicates if address is a commercial mail-receiving agent (CMRA).
FLD_DPV_IS_NOSTAT	Out	String	Indicates if the address is not receiving delivery and is not counted as a possible delivery. The address is not receiving delivery because: 1) delivery has not been established, 2) The customer receives mail as part of a drop, 3) the carrier destroys or returns all of the mail.
FLD_DPV_IS_PBSA	Out	Long	Indicates if address is a Post Office Box street address (PBSA).
FLD_DPV_IS_THROWBACK	Out	Boolean	Indicates if the address associated with the delivery point is a street address; however, the delivery is made to the customer's PO Box address.
FLD_DPV_IS_VACANT	Out	String	Indicates if address is unoccupied.
FLD_DPV_NO_SECURE_LOCATION	Out	Boolean	Indicates if the location of the address is not secure. The USPS can access the door, but cannot leave a package due to security concerns.
FLD_DROP_SITE_KEY	Out	String	Returns the value of the unique key for the drop site for mailpieces that are drop shipped. If the piece is delivered to the origin point, returns "ORIGIN".
FLD_ENDORSEMENT_LINE	Out	String	Endorsement line.
FLD_ERRORCODE	In/Out	Long	Error code returned from address correction process.

Field ID	Data Direction	Data Type Returned	Description
FLD_EXTRA_INFO	In/Out	String	Extra information.
FLD_FIRM_BUNDLE_COPIES	In/Out	Long	The mail piece count to be delivered to this address. Records sorted into a firm bundle will indicate their position in the count, not the total number of copies to be sent to this address.
FLD_FIRM_BUNDLE_INDICATOR	Out	String	Indicates whether this address is a master record or a duplicated record to account for multiple copies sent to a single address. Is one of the following: <ul style="list-style-type: none"><li>• C – This is a copied address.</li><li>• M – This is a master address that has produced copied addresses.</li><li>• blank – Other address.</li></ul>
FLD_FIRST_NAME	In/Out	String	First name.
FLD_FULL_NAME	In/Out	String	The full name of the addressee.
FLD_GEOCODE_CENSUS_BLOCK	Out	String	Census block – requires Geocode add-on.
FLD_GEOCODE_CENSUS_TRACT	Out	String	Census tract – requires Geocode add-on.

Field ID	Data Direction	Data Type Returned	Description
FLD_GEOCODE_ FOOTNOTE	Out	String	Code indicating the granularity of ZIP Code match (ZIP Code length) for a geocode lookup – requires Geocode add-on: <ul style="list-style-type: none"> <li>• 00 - Unable to perform a Geocode lookup.</li> <li>• 03 - Geocode data based on a 3-digit ZIP Code.</li> <li>• 05 - Geocode data based on a 5-digit ZIP Code.</li> <li>• 07 - Geocode data based on a 7-digit ZIP Code.</li> <li>• 09 - Geocode data based on a 9-digit ZIP Code.</li> </ul>
FLD_GEOCODE_ LATITUDE	Out	String	Latitude coordinate – requires Geocode add-on.
FLD_GEOCODE_ LONGITUDE	Out	String	Longitude coordinate – requires Geocode add-on.
FLD_GEOCODE_ MSA_CODE	Out	String	Metropolitan statistical area code – requires Geocode add-on.
FLD_IM_BARCODE	Out	String	Intelligent Mail barcode (presort field only); requires that the usps4cb.ttf font be installed.
FLD_IM_BARCODE_ NUMERIC	Out	String	Human-readable format of the Intelligent Mail barcode (presort field only).

Field ID	Data Direction	Data Type Returned	Description
FLD_IM_CONTAINER_BARCODE	Out	String	Contains the Intelligent Mail barcode used in the tray or sack that contains this mail piece.
FLD_IM_PACKAGE_BARCODE	Out	String	Data needed for Intelligent Mail package barcode (IMpb); requires that the SatIMpb font (72 point) be installed
FLD_IM_PACKAGE_BARCODE_HUMAN	Out	String	Data needed for human readable element above IMpb barcode
FLD_IM_PACKAGE_BARCODE_NUMERIC	Out	String	Data needed for numeric element below IMpb barcode
FLD_IM_PALLET_BARCODE	Out	String	Contains the Intelligent Mail barcode used in the pallet that contains this mail piece.
FLD_IM_PIECE_DATA_INPUTS	In/Out	String	Stores third-party IMb tracking information. Can include Barcode ID, Service Type  ID (STID), Mailer ID (MID), Sequence Number, and/or Routing number used in the mailing.
FLD_IM_PIECE_IDENTIFIER	In/Out	String	An optional custom number that can be used by a Track N Trace mailer to identify mail pieces or groups of mail pieces on the Track N Trace Web Portal. The number can be u Distinct from FLD_IM_PIECE_SEQUENCE_ID.

Field ID	Data Direction	Data Type Returned	Description
FLD_IM_PIECE_SEQUENCE_ID	In/Out	Long	Contains the sequence number used in this mail piece's Intelligent Mail barcode. Set this field when you set the property ptIM_SEQUENCING_METHOD to 3, use input field.
FLD_IS_NONSUBSCRIBER	In/Out	Long	Indicates Periodicals copy is being sent to a non-subscriber
FLD_IS_RESIDENCE	In/Out	Long	Indicates if address is residential. Returned from CASS processing if you are using the RDI add-in. Can be used for Presort processing to qualify Every Door Direct Mail® (EDDM) mailings.
FLD_KEYLINE	Out	String	Keyline to be printed on address labels for mixed-weight First Class Mail sorts (Manifesting).
FLD_LACS_FOOTNOTE	Out	String	Code indicating specific information about a LACS <sup>Link</sup> lookup: <ul style="list-style-type: none"> <li>• "" - Not processed / Seed record.</li> <li>• 00 - No match.</li> <li>• 09 - Matched to default high-rise address; address not updated.</li> <li>• 14 - Match failure to build new address.</li> <li>• 92 - Match secondary dropped from input.</li> <li>• A - Match success.</li> </ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_LACS_INDICATOR	Out	String	<p>A single character indicating the result of a LACS<sup>Link</sup> lookup:</p> <ul style="list-style-type: none"> <li>• "" - Not processed.</li> <li>• N - Match / Matched, but there was a failure to build new address.</li> <li>• Y - Match success.</li> <li>• S - Match with secondary dropped from input.</li> <li>• F - Seed record.</li> </ul>
FLD_LAST_NAME	In/Out	String	Last name.
FLD_LASTLINE	In/Out	String	Field containing city, state and ZIP Code; required for CASS processing unless FLD_CITY, FLD_STATE and FLD_ZIPCODE already specified.
FLD_LONG_ERROR_STRING	Out	String	Extended description of the error code.
FLD_LOT_NUMBER	In/Out	String	Line-of-Travel number.

Field ID	Data Direction	Data Type Returned	Description
FLD_MATCH_FLAG	Out	String	Match flag returned from NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"> <li>• M – Matched; updated address.</li> <li>• F – Foreign move; new address unavailable.</li> <li>• K – No forwarding address; new address unavailable.</li> <li>• G – PO box closed; new address unavailable.</li> <li>• N – No match.</li> <li>• X – Other.</li> </ul>
FLD_MATCHED_TO_DEFAULT	Out	BOOL	TRUE if record should contain additional secondary address information (but is not required to).
FLD_MAILDAT_INFORMED_OPTIN	Out	Bool	Indicates that the mailing is a campaign in the USPS Informed Delivery <sup>®</sup> program.
FLD_MAILDAT_INFORMED_REPRESENT	In/Out	String	URL pointer to an image to replace the default scanned grayscale image of the mail piece.
FLD_MAILDAT_INFORMED_RIDEALONG	In/Out	String	URL pointer to an image that includes a clear call to action or next steps.
FLD_MAILDAT_INFORMED_TARGETURL	In/Out	String	URL pointer to a website or portal to provide a customer with more information.

Field ID	Data Direction	Data Type Returned	Description
FLD_MOVE_EFFECTIVE	Out	String	Move effective date returned from NCOA <sup>Link</sup> .
FLD_MOVE_FOOTNOTE	Out	String	<p>NCOA footnote code:</p> <p><b>Match Found – new address returned:</b></p> <hr/> <p>A            Input record matched.</p> <p>91           Secondary number dropped from change of address.</p> <p>92           Secondary number dropped from input address.</p> <p><b>Match Found – new address unavailable:</b></p> <hr/> <p>1            Foreign move.</p> <p>2            Move left no address.</p> <p>3            PO box closed; no forwarding.</p> <p>5            New 11-digit DPBC is ambiguous.</p> <p>14           New address would not convert to deliverable</p> <p>19           ZIP Code not found, or Temporary Change of Address.</p>

Field ID	Data Direction	Data Type Returned	Description
			<p><b>No Match Found:</b></p> <hr/> <p>00 No move found.</p> <p>4 Street address missing secondary.</p> <p>6 Conflicting directions, middle name related.</p> <p>7 Conflicting directions, gender related.</p> <p>8 Other conflicting instructions.</p> <p>9 High-rise default.</p> <p>10 Rural route default.</p> <p>11 Individual, insufficient name for match.</p> <p>12 Middle name test failed.</p> <p>13 Gender test failed.</p> <p>15 Individual name insufficient.</p> <p>16 Secondary number discrepancy.</p> <p>17 Other insufficient name.</p> <p>18 General delivery.</p> <p>20 Conflicting directions after re-chaining.</p> <p>66 Address deleted, no forwarding allowed.</p>
FLD_MOVE_FOOTNOTE_LONG_DESCRIPTION	Out	String	NCOA <sup>Link</sup> long footnote description.
FLD_MOVE_FOOTNOTE_SHORT_DESCRIPTION	Out	String	NCOA <sup>Link</sup> short footnote description.

Field ID	Data Direction	Data Type Returned	Description
FLD_MOVE_TYPE	Out	String	Move type returned from NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"> <li>• I – Individual</li> <li>• F – Family</li> <li>• B – Business</li> </ul>
FLD_NCOA_ADDRESSLINE1	Out	String	Address Line 1 after NCOA <sup>Link</sup> processing.
FLD_NCOA_ADDRESSLINE2	Out	String	Address Line 2 after NCOA <sup>Link</sup> processing.
FLD_NCOA_CARRIER_ROUTE	Out	String	Carrier route after NCOA <sup>Link</sup> processing.
FLD_NCOA_CITY	Out	String	City after NCOA <sup>Link</sup> processing.
FLD_NCOA_CITY_ABBREVIATED	Out	String	Abbreviated city after NCOA <sup>Link</sup> processing.
FLD_NCOA_COUNTY_CODE	Out	String	County code after NCOA <sup>Link</sup> processing.
FLD_NCOA_COUNTY_NAME	Out	String	County name after NCOA <sup>Link</sup> processing.
FLD_NCOA_DPBARCODE	Out	String	Delivery point barcode string after NCOA <sup>Link</sup> processing.
FLD_NCOA_DPC	Out	String	Delivery Point/Check Digit after NCOA <sup>Link</sup> processing.
FLD_NCOA_DPV_CODED	Out	String	Indicates if the address, after NCOA <sup>Link</sup> processing, was confirmed with DPV.

Field ID	Data Direction	Data Type Returned	Description
FLD_NCOA_DPV_DOOR_NOT_ACCESSIBLE	Out	Boolean	Indicates if the address, after NCOALink processing, is not physically accessible by carriers.
FLD_NCOA_DPV_FOOTNOTE	Out	String	<p>After NCOA<sup>Link</sup> processing, the DPV footnote code:</p> <ul style="list-style-type: none"> <li>LK Processing locked out due to a seed record being processed.</li> <li>AA Matched to the ZIP+4 file</li> <li>A1 No match against the ZIP+4 file</li> <li>BB Matched to DPV file (all components confirmed)</li> <li>CC Matched only after removing secondary information; they were presented but invalid.</li> <li>N1 Input primary matched, but high-rise missing secondary number.</li> <li>M1 Primary number missing.</li> <li>M3 Primary number invalid.</li> <li>P1 Input missing PO, RR or HC box number.</li> <li>P3 Failed DPV because of invalid PO, RR or HC box number.</li> <li>RR Matched CMRA (found in CMRA file).</li> <li>R1 Matched CMRA, but secondary number (i.e., PMB) missing.</li> <li>U1 Matched unique ZIP Code.</li> <li>G1 Matched general delivery.</li> <li>F1 Matched military address.</li> </ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_NCOA_DPV_INDICATOR	Out	String	<p>After NCOA<sup>Link</sup> processing, returns a single character that indicates the level of address validity with DPV:</p> <ul style="list-style-type: none"><li>• Y - Both the primary and secondary (if present) validated against the DPV database.</li><li>• S - The primary address is valid according to DPV, but the secondary is invalid.</li><li>• D - The primary address is valid according to DPV, but the address is missing secondary information.</li><li>• N - The primary address is not valid according to DPV.</li><li>• "" - The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.</li><li>• X - The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.</li><li>• E - The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.</li></ul>

Field ID	Data Direction	Data Type Returned	Description
FLD_NCOA_DPV_IS_CMRA	Out	String	Indicates if the address, after NCOA <sup>Link</sup> processing, is a commercial mail-receiving agent (CMRA).
FLD_NCOA_DPV_IS_NOSTAT	Out	String	Indicates if the address, after NCOA <sup>Link</sup> processing, is not receiving delivery and is not counted as a possible delivery. The address is not receiving delivery because: 1) delivery has not been established, 2) The customer receives mail as part of a drop, 3) the carrier destroys or returns all of the mail.
FLD_NCOA_DPV_IS_PBSA	Out	String	Indicates if the address, after NCOA <sup>Link</sup> processing, is a Post Office Box street address (PBSA).
FLD_NCOA_DPV_IS_THROWBACK	Out	Boolean	Indicates if the address, after NCOA <sup>Link</sup> processing, is a street address for which delivery is made to the customer's PO Box address.
FLD_NCOA_DPV_IS_VACANT	Out	String	Indicates if the address, after NCOA <sup>Link</sup> processing, is unoccupied.
FLD_NCOA_DPV_NO_SECURE_LOCATION	Out	Boolean	Indicates if the address, after NCOA <sup>Link</sup> processing, is not a secure location.
FLD_NCOA_DUAL_ADDRESS_LINE	In/Out	String	Used by the Job Files add-on for BCC Architect. When more than one address is submitted, stores the address that was submitted but not used for address correction -- (e.g., the PO Box when the street address was used for address correction)

Field ID	Data Direction	Data Type Returned	Description
FLD_NCOA_ERROR_CODE	Out	String	Error code after NCOA <sup>Link</sup> processing.
FLD_NCOA_LOT_NUMBER	Out	String	LOT number after NCOA <sup>Link</sup> processing.
FLD_NCOA_MATCH_FIRST_NAME	Out	String	First name of the person used for NCOA <sup>Link</sup> matching.
FLD_NCOA_MATCH_LAST_NAME	Out	String	Last name of the person used for NCOA <sup>Link</sup> matching.
FLD_NCOA_MATCH_SUFFIX_NAME	Out	String	Suffix of the person used for NCOA <sup>Link</sup> matching.
FLD_NCOA_MIDDLE_INITIAL	Out	String	Middle initial of the person used for NCOA <sup>Link</sup> matching.
FLD_NCOA_PO_BOX_DELIVERY_ONLY_ZIP	Out	String	Indicates whether delivery is to an address in a PO Box only zone, after NCOA <sup>Link</sup> processing
FLD_NCOA_POST_DIRECTIONAL	Out	String	Post-directional after NCOA <sup>Link</sup> processing.
FLD_NCOA_PRE_DIRECTIONAL	Out	String	Predirectional after NCOA <sup>Link</sup> processing.
FLD_NCOA_PRIMARY_ADDRESS_LINE	In/Out	String	Used by the Job Files add-on for BCC Architect. The primary address line information.
FLD_NCOA_PRIMARY_NUMBER	Out	String	Primary number after NCOA <sup>Link</sup> processing.

Field ID	Data Direction	Data Type Returned	Description
FLD_NCOA_RECORD_TYPE	Out	String	Record type after NCOA <sup>Link</sup> processing.
FLD_NCOA_SECONDARY_ADDRESS_LINE	In/Out	String	Used by the Job Files add-on for BCC Architect. The secondary information (unit designator / unit information) for the address.
FLD_NCOA_STATE	Out	String	State after NCOA <sup>Link</sup> processing.
FLD_NCOA_STREET_NAME	Out	String	Street name after NCOA <sup>Link</sup> processing.
FLD_NCOA_SUFFIX	Out	String	Suffix after NCOA <sup>Link</sup> processing.
FLD_NCOA_UNIT_DESIGNATOR	Out	String	Unit designator after NCOA <sup>Link</sup> processing.
FLD_NCOA_UNIT_NUMBER	Out	String	Unit number after NCOA <sup>Link</sup> processing.
FLD_NCOA_URBANIZATION	Out	String	Urbanization after NCOA <sup>Link</sup> processing. Puerto Rico only.
FLD_NCOA_VANITY_ADDRESS_BLOCK	In/Out	String	Used by the Job Files add-on for BCC Architect. An alternate version of the address block which preserves the originally submitted sequence of address elements.
FLD_NCOA_ZIPCODE	Out	String	ZIP Code after NCOA <sup>Link</sup> processing.
FLD_ORIG_ADDRESSLINE1	Out	String	Address Line 1 of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_ADDRESSLINE2	Out	String	Address Line 2 of the input address sent to NCOA <sup>Link</sup> processing.

Field ID	Data Direction	Data Type Returned	Description
FLD_ORIG_BUSINESS	Out	String	Business name of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_CARRIER_ROUTE	Out	String	Carrier route of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_CITY	Out	String	City of the input address sent to NCOA <sup>Link</sup> processing
FLD_ORIG_DPC	Out	String	Delivery Point/Check Digit for the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_FIRSTNAME	Out	String	First name of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_FULL_NAME	Out	String	Full name of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_LASTLINE	Out	String	Last line of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_LASTNAME	Out	String	Last name of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_MIDDLENAME	Out	String	Middle name of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_POST_DIRECTIONAL	Out	String	Postdirectional of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_PRE_DIRECTIONAL	Out	String	Predirectional of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_PREFIXTITLE	Out	String	Prefix title of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_PRIMARY_NUMBER	Out	String	Primary number of the input address sent to NCOA <sup>Link</sup> processing.

Field ID	Data Direction	Data Type Returned	Description
FLD_ORIG_STATE	Out	String	State for the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_STREET_NAME	Out	String	Street name of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_SUFFIX	Out	String	Suffix of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_SUFFIXTITLE	Out	String	Suffix title of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_UNIT_DESIGNATOR	Out	String	Unit designator of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_UNIT_NUMBER	Out	String	Unit number of the input address sent to NCOA <sup>Link</sup> processing.
FLD_ORIG_URBANIZATION	Out	String	Urbanization of the input address sent to NCOA <sup>Link</sup> processing. Puerto Rico only.
FLD_ORIG_ZIPCODE	Out	String	ZIP Code of the input address sent to NCOA <sup>Link</sup> processing.
FLD_PACKAGE_NUMBER	Out	Long	The bundle (formerly referred to as the package) number.
FLD_PALLET_DESTINATION	Out	String	Pallet destination.
FLD_PALLET_LEVEL	Out	String	Pallet level.
FLD_PALLET_NUMBER	Out	Long	Pallet number.

Field ID	Data Direction	Data Type Returned	Description
FLD_PIECE_ENTRY_POINT_ZIPCODE	Out	String	Returns the ZIP Code and description (if you use PVDS) of the SCF, NDC or ASF where this mail piece is to be deposited.
FLD_PIECE_INCOUNTY	Out	Long	Indicates if address is in-county. For Periodicals sorts only.
FLD_PIECE_POSTAGE	Out	String	Exact postage per piece.
FLD_PIECE_THICKNESS	In/Out	String	Piece thickness; used for mixed-weight First Class Mail sorts (Manifesting).
FLD_PIECE_WEIGHT	In/Out	String	Piece weight in ounces; used for mixed-weight First Class Mail sorts (Manifesting).
FLD_PIECE_ZONENUMBER	Out	String	Zone number of address for a Periodicals sort.
FLD_PMB_NUMBER	Out	String	Private mailbox number.
FLD_PO_BOX_DELIVERY_ONLY_ZIP	Out	String	Indicates whether delivery is to an address in a PO Box only zone.
FLD_POST_DIRECTIONAL	Out	String	Post-directional address element.
FLD_PRE_DIRECTIONAL	Out	String	Pre-directional address element.
FLD_PRESORT_ID	Out	Long	The rank order of a mail piece in a particular sort.

Field ID	Data Direction	Data Type Returned	Description
FLD_PRIMARY_NUMBER	Out	String	Primary number address element.
FLD_RECORD_ID	In/Out	String	User field that can contain the input record's index or ID.
FLD_RECORD_TYPE	Out	String	The following record types are currently valid: <ul style="list-style-type: none"> <li>• S – Street record</li> <li>• P – Post office box</li> <li>• R – Rural Route or Highway Contract</li> <li>• H – High-rise, Building or Apartment</li> <li>• F – Firm Record</li> <li>• G – General Delivery</li> </ul>
FLD_SHORT_ERROR_STRING	Out	String	Standard description of error code.
FLD_SKIPPED_CERTIFY	Out	BOOL	TRUE if record was skipped during CASS processing.
FLD_STATE	In/Out	String	State; required for CASS processing unless FLD_LASTLINE specified.
FLD_STREET_NAME	Out	String	Street name address element.
FLD_SUBSCRIBER_FLAG	In/Out	BOOL	Not currently implemented.
FLD_SUFFIX	Out	String	Suffix address element.

Field ID	Data Direction	Data Type Returned	Description
FLD_SUITELINK_FOOTNOTE	Out	String	<p>Suite<sup>Link</sup> footnote code:</p> <ul style="list-style-type: none"> <li>• "" – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup within the Suite<sup>Link</sup> file. Only default high-rise addresses qualify for a Suite<sup>Link</sup> lookup.</li> <li>• A – The address was processed and secondary information was added to the resulting address.</li> <li>• 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.</li> </ul>
FLD_TRAY_DESTINATION	Out	String	Not currently implemented.
FLD_TRAY_NUMBER	Out	Long	Tray or sack number.
FLD_TRAY_TYPE	Out	String	Tray type.
FLD_UNIT_DESIGNATOR	Out	String	Unit designator address element.
FLD_UNIT_NUMBER	Out	String	Unit number address element.
FLD_URBANIZATION	Out	String	Urbanization number. Puerto Rico addresses only.
FLD_USER_DEFINED_1	In/Out	String	Not currently implemented; will only be used when printing labels.

Field ID	Data Direction	Data Type Returned	Description
FLD_USER_DEFINED_2	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_3	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_4	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_5	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_6	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_7	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_8	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_9	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_10	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_11	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_12	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_13	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_USER_DEFINED_14	In/Out	String	Not currently implemented; will only be used when printing labels.

Field ID	Data Direction	Data Type Returned	Description
FLD_USER_DEFINED_15	In/Out	String	Not currently implemented; will only be used when printing labels.
FLD_WALK_SEQUENCE_NUMBER	In/Out	String	Walk Sequence number.
FLD_ZIP4_FOOTNOTE	Out	String	ZIP4 footnote code.
FLD_ZIPCODE	In/Out	String	ZIP Code; required for CASS processing unless FLD_LASTLINE specified.

## COM MRTKTaskLib Result Codes

Below are the result codes currently returned from each task. They indicate the cause of an error encountered when performing a specific Task's operations. The codes are associated with a string that gives a more detailed explanation of the error. The VB Err object will contain the COM MRTKTaskLib result code if an error occurs in a Visual Basic project.

Result Code	Hex Value	Description
E_REPORT_XML_LOAD_FAILURE	0x80003333	A report section couldn't be loaded from the source XML.
E_REPORT_XML_ADD_FAILURE	0x80003334	A failure occurred while adding a report to the blob.
E_REPORT_XML_NOT_INSTALLED_PROPERLY	0x80003335	MSXML failed to instantiate.
E_REPORT_XML_NOT_INITIALIZED	0x80003336	The report blob wasn't completely initialized, but was used anyway.
E_REPORT_REPORT_ENGINE_NOT_INITIALIZED	0x80003337	The printing engine wasn't completely initialized, but was used anyway.

Result Code	Hex Value	Description
E_REPORT_UNKNOWN_REPORT_ID	0x80003338	The printing engine didn't recognize the report ID requested.
E_REPORT_REPORT_NOT_AVAILABLE	0x80003339	No reports were specified, or a report couldn't be loaded.
E_REPORT_REPORT_ERROR_READING	0x80003340	A report couldn't be loaded from a resource.
E_REPORT_GENERIC_FILE_SAVE_ERROR	0x80003341	The temporary report file couldn't be created.
E_REPORT_XML_DATA_OBJECT_NOT_SET	0x80003342	The report blob hasn't yet been set up.
E_REPORT_REPORT_BAD_PARAMETER	0x80003343	Invalid argument passed to the printing engine.
E_REPORT_PROPERTIES_OBJECT_NOT_SET	0x80003344	The printing engine's properties were not set.
E_REPORT_ACTIVE_REPORTS_NOT_INSTALLED	0x80003345	Active Reports failed to instantiate.
E_REPORT_PREVIEW_WINDOW_NOT_CREATED	0x80003346	The Active Reports OCX failed to instantiate.
E_REPORT_NO_REPORT_LOADED	0x80003347	Unable to load a report.
E_REPORT_INVALID_HINSTANCE	0x80003348	No DLL was specified (for loading report resources).
E_REPORT_SOME_REPORTS_FAILED_LOAD	0x80003349	Unable to load one or more reports.
E_REPORT_REPORT_FAILED_PRINT	0x80003350	Printing a report failed.

Result Code	Hex Value	Description
E_MRTK_DPV_EXPIRED	0x80040491L	Indicates the DPV add-on registration number has expired. THIS CODE IS OBSOLETE.
E_MRTK_DPV_LOCKED_OUT	0x80040490L	Indicates that DPV has been locked out after attempting to validate a seed record. Contact BCC Software for an unlock key.
E_MRTK_DPV_NOT_ALLOWED_EXPORTED	0x80040492	Indicates DPV processing failed because the address-matching engine does not reside in the US.
E_MRTK_ENGINE_NOT_INITIALIZED	0x80040403L	Indicates that the CASS matching engine could not be initialized.
E_MRTK_EXPIRED	0x80040413L	Indicates that a component of BCC Architect has expired. For instance, CASSTask may have expired, so a call to CASSTask.PrepareTask will return this error.
E_MRTK_FAILED_TEMP_FILE	0x8004041aL	Indicates that one of several temporary file managers failed during their construction, initialization, etc.
E_MRTK_GENERAL_FAIL	0x80040400L	Indicates a general error such as an inability to initialize the CASSTask object.
E_MRTK_GEOCODE_DATAFILE_EXPIRED	0x800404c8	Indicates the Geocode.cas data file has expired.
E_MRTK_GEOCODE_DATAFILE_FORMAT_INCORRECT	0x800404c7	Indicates the Geocode.cas data file is not in the correct format.
E_MRTK_GEOCODE_DATAFILE_NEW	0x800404ca	Indicates the Geocode.cas data file is newer than the Address.cas file.

Result Code	Hex Value	Description
E_MRTK_GEOCODE_DATAFILE_NOT_FOUND	0x800404c6	Indicates the Geocode.cas data file could not be found.
E_MRTK_GEOCODE_DATAFILE_OLD	0x800404c9	Indicates the Geocode.cas data file is older than the Address.cas file.
E_MRTK_ILLEGAL_TYPE_CONVERSION	0x800404A0L	Indicates the argument supplied for conversion is not of the correct data type.
E_MRTK_IN_FIELD_CNT_HIGH	0x80040442L	BCC Architect Server Specific: Indicates that the input field count is high for the specified request type.
E_MRTK_IN_FIELD_CNT_LOW	0x80040441L	BCC Architect Server Specific: Indicates that the input field count is too low for the specified request type.
E_MRTK_INCORRECT_INPUT_FIELDMAP	0x8004040aL	Indicates that some fields required for CASS processing were not mapped. For example, AddressLine1 or AddressLine2 are not mapped. See the Field IDs table for details as to which fields are required.
E_MRTK_INTERFACE_FAILED	0x80040409L	Indicates that an interface used by a given task failed. The ZIPTask.PrepareTask function could return this error if the CASS Engine interface failed to be created.
E_MRTK_INTERFACE_NOT_AVAILABLE	0x80040405L	Indicates that an interface cannot be created or used. This can occur if a user has the BCC Architect Office product, but they attempt to use Architect features such as Pre-sortTask.

Result Code	Hex Value	Description
E_MRTK_INVALID_ARG	0x80040408L	Indicates that a function argument was not in the correct range, etc. For instance, calling Pre-sortTask.Retrieve after setting the number of records to retrieve to a negative number.
E_MRTK_INVALID_CALL_SEQUENCE	0x8004040cL	Indicates that the sequence of function calls in a particular task was not correct. For instance, PrepareTask and ValidateProperties must be called before the CASSTask.Print3553 function.
E_MRTK_INVALID_INPUT_BLOCK	0x8004040bL	Indicates that the input block is formatted incorrectly. This can occur if an empty address block is passed to the CASSTask.Update function.
E_MRTK_INVALID_REG_KEY	0x8004041bL	Indicates that the current registration key is invalid.
E_MRTK_LACS_DATAFILE_EXPIRED	0x800404d7	Indicates the LACSLink.cas data file has expired.
E_MRTK_LACS_DATAFILE_FORMAT_INCORRECT	0x800404d6	Indicates the LACSLink.cas data file is not in the correct format.
E_MRTK_LACS_DATAFILE_NEW	0x800404d9	Indicates the LACSLink.cas data file is newer than the Address.cas file.
E_MRTK_LACS_DATAFILE_NOT_FOUND	0x800404d5	Indicates the LACSLink.cas data file could not be found.
E_MRTK_LACS_DATAFILE_OLD	0x800404d8	Indicates the LACSLink.cas data file is older than the Address.cas file.

Result Code	Hex Value	Description
E_MRTK_LACS_NOT_ALLOWED_EXPORTED	0x80040493	Indicates LACS <sup>Link</sup> processing failed because the address-matching engine does not reside in the US.
E_MRTK_LACSLINK_LOCKED_OUT	0x800404df	Indicates that LACS <sup>Link</sup> processing has been locked out because a seed record has been encountered.
E_MRTK_LOT_DATAFILE_EXPIRED	0x800404dc	Indicates the eLOT.cas data file has expired.
E_MRTK_LOT_DATAFILE_FORMAT_INCORRECT	0x800404db	Indicates the eLOT.cas data file is not in the correct format.
E_MRTK_LOT_DATAFILE_NEW	0x800404de	Indicates the eLOT.cas data file is newer than the Address.cas file.
E_MRTK_LOT_DATAFILE_NOT_FOUND	0x800404da	Indicates the eLOT.cas data file could not be found.
E_MRTK_LOT_DATAFILE_OLD	0x800404dd	Indicates the eLOT.cas data file is older than the Address.cas file.
E_MRTK_MATCHING_LOAD_FAILED	0x80040402L	Indicates that the CASS Matching Engine could not be loaded.
E_MRTK_MAX_SEARCH_COUNT	0x800404B7L	Indicates that the number of calls to USZIPTask.NextAddress exceeded SearchCount.
E_MRTK_NOT_IMPLEMENTED	0x80040415L	Indicates that a specific feature, function or interface has not been implemented.
E_MRTK_NOT_IN_BATCH_MODE	0x80040401L	Indicates that a batch record process was attempted when not in batch mode.

Result Code	Hex Value	Description
E_MRTK_NULL_ARG	0x80040407L	Indicates that a NULL was used as an argument to a function that requires non-NULL arguments. For example, calling the Pre-sortTask.Send function with a NULL argument.
E_MRTK_OUT_FIELD_CNT_HIGH	0x80040444L	BCC Architect Server Specific: Indicates that the output field count is too high for the specified request type.
E_MRTK_OUT_FIELD_CNT_LOW	0x80040443L	BCC Architect Server Specific: Indicates that the output field count is too low for the specified request type.
E_MRTK_OUT_OF_MEMORY	0x80040406L	Indicates that not enough memory was available to allocate an object or interface.
E_MRTK_PRINTER_ERROR	0x80040416L	Indicates a general printer error such as attempting to print to a printer that does not exist.
E_MRTK_RDI_DATAFILE_EXPIRED	0x800404c3	Indicates the RDI data files have expired.
E_MRTK_RDI_DATAFILE_FORMAT_INCORRECT	0x800404c2	Indicates the RDI data files are not in the correct format.
E_MRTK_RDI_DATAFILE_NEW	0x800404c5	Indicates the RDI data files are newer than the Address.cas file.
E_MRTK_RDI_DATAFILE_NOT_FOUND	0x800404c1	Indicates the RDI data files could not be found.
E_MRTK_RDI_DATAFILE_OLD	0x800404c4	Indicates the RDI data files are older than the Address.cas file.

Result Code	Hex Value	Description
E_MRTK_REG_KEY_ACCESS_DENIED	0x8004041cL	Indicates the registration key could not be created or read.
E_MRTK_REGISTRATION_KEY_INVALID	0x80040448L	BCC Architect Server Specific: Indicates the registration key specified within the request is invalid.
E_MRTK_REQUEST_FORMAT_INVALID	0x80040447L	BCC Architect Server Specific: Indicates that the request format is invalid.
E_MRTK_REQUEST_KEY_INVALID	0x80040440L	BCC Architect Server Specific: Indicates that the request key is invalid.
E_MRTK_REQUEST_LEN_INVALID	0x80040446L	BCC Architect Server Specific: Indicates that the size specified within the request does not match the actual input size.
E_MRTK_REQUEST_TOO_LARGE	0x80040445L	BCC Architect Server Specific: Indicates that the request is over 1024K.
E_MRTK_SERVER_BUSY	0x80040452L	BCC Architect Server Specific: Indicates an attempt to shut down the server failed because it is processing a job.
E_MRTK_SERVER_ERROR	0x80040450L	BCC Architect Server Specific: Indicates general error.
E_MRTK_SERVER_INVALID	0x8004044aL	BCC Architect Server Specific: Indicates that the BCC Architect Server specified is incorrect. Make sure that the format is of the type "Server-NameOrIP: PortNumber."

Result Code	Hex Value	Description
E_MRTK_SERVER_OFFLINE	0x80040454L	Indicates that the BCC Architect Server client was unable to connect to the server because it is off-line.
E_MRTK_SERVER_SHUTDOWN	0x80040451L	Indicates that the BCC Architect Server client was unable to connect to the server because it is shut down.
E_MRTK_SERVER_TIME_OUT	0x80040455L	BCC Architect Server Specific: Indicates an attempt to contact the server timed out.
E_MRTK_SORT_DENSITY_NOT_MET	0x8004041DI	Indicates that the density required for a particular sort was not satisfied.
E_MRTK_SORT_GENERIC_FAILURE	0x8004041eL	Indicates an error occurred during presorting.
E_MRTK_SORT_NOT_ENOUGH_IDS_TO_ALLOCATE	0x80040421L	Indicates unable to allocate any sequence numbers because there are not enough available. One scenario in which this may happen is that of a large volume mailer using a 9-digit mailer ID.
E_MRTK_SORT_NOT_ENOUGH_PIECES	0x80040419L	Indicates that the minimum number of pieces required for a particular sort was not met.
E_MRTK_SORT_PRESORTED_LIMIT_EXCEEDED	0x80040420	Indicates a specified minimum or maximum cannot be achieved in a sort (currently, a maximum of 10% of Standard Mail flats prepared in trays may be claimed at Presorted rates).

Result Code	Hex Value	Description
E_MRTK_SORT_SAVE_FILES_FAILED	0x80040429L	Indicates that electronic documentation, for example, Mail.dat or .ssf file, cannot be generated with the specified sort settings. No files will be saved.
E_MRTK_SORT_SEQUENCE_NUMBER_INVALID	0x80040423L	Indicates that the Intelligent Mail sequence number is a negative number. This number must be greater than zero.
E_MRTK_SORT_SEQUENCE_NUMBER_OUT_OF_RANGE	0x80040422L	Indicates that the sequence number is too large. If you have a six-digit Mailer ID, your sequence number must have no more than nine digits. If you have a nine-digit Mailer ID, your sequence number must have no more than six digits.
E_MRTK_SUCCESS	0	Task was successfully completed.
E_MRTK_SUITELINK_DATAFILE_EXPIRED	0x800404e3	Indicates the SuiteLink.cas data file has expired.
E_MRTK_SUITELINK_DATAFILE_FORMAT_INCORRECT	0x800404e2	Indicates the SuiteLink.cas data file is not in the correct format.
E_MRTK_SUITELINK_DATAFILE_NEW	0x800404e5	Indicates the SuiteLink.cas data file is newer than the Address.cas file.
E_MRTK_SUITELINK_DATAFILE_NOT_FOUND	0x800404e1	Indicates the SuiteLink.cas data file could not be found.
E_MRTK_SUITELINK_DATAFILE_OLD	0x800404e4	Indicates the SuiteLink.cas data file is older than the Address.cas file.

Result Code	Hex Value	Description
E_MRTK_SYSTEM_COMM_ERROR	0x8004044bL	BCC Architect Server Specific: Indicates that the client was unable to load the Winsock communication libraries or that the local network is not set up correctly.
E_MRTK_TEMPLATE_ALREADY_EXISTS	0x800404A3L	Indicates the specified presort template already exists.
E_MRTK_TEMPLATE_DOES_NOT_EXIST	0x800404A2L	Indicates the specified presort template does not exist.
E_MRTK_TEMPLATE_LOCKED	0x800404A4L	Indicates the specified presort template could not be modified because it is in use.
E_MRTK_TEMPLATE_UNLOCKED	0x800404A5L	Indicates the specified presort template is no longer locked.
E_MRTK_UPDATE_FAILED	0x80040418L	Indicates that the AutoUpdater failed to update BCC Architect
E_MRTK_UPDATE_FAILED_DISKFULL	0x80040480L	Indicates the AutoUpdater failed because disk was full.
E_MRTK_UPDATE_RESTART_NEEDED	0x80040417L	Indicates that a system reboot is required to complete the BCC Architect update.
E_MRTK_USER_CANCELED	0x80040414L	Indicates that "Cancel" was hit during a process. For instance, PresortTask.Send will return this error if the user hits the "Cancel" button. This assumes that graphical user elements have been enabled.
E_MRTK_USER_RESTART	0x800404A1L	Indicates that a restart is required.

# .NET Classes Reference

## Contents

The .NET ZIPAssembly Class for Correcting Single Addresses .....	516
Using ZIPAssembly for Single Address Correction .....	517
ZIPAssembly Functions .....	517
BrowseAddress .....	518
BuildAddress .....	519
CheckAddress .....	520
ClearAddress .....	521
EndTask .....	521
ParseAddress .....	522
PrepareTask .....	523
.NET ZIPAssembly Properties .....	524
AddressBlock .....	524
AddressInputPreference .....	527
AddressLine1 .....	528
AddressLine2 .....	529
AddressLineAbbreviated .....	530
AddressSuggestionList .....	531
ApplyCasingBusiness .....	532
AssignLOT .....	532
AssignRDI .....	533
BusinessName .....	533
CarrierRoute .....	534
CarrtCoded .....	534
Casing .....	535
CASSDate .....	535
CensusBlock .....	536
CensusTract .....	537
City .....	537
CityFormat .....	538
CityStateZip .....	539
CongressionalDistrict .....	539
CountyCode .....	540
CountyFIPSCode .....	540
CountyName .....	540
DeliveryPointCheckDigit .....	541
BarcodeString .....	541
DPVCoded .....	542
DPVFailureAsError .....	542
DPVFootnotes .....	543
DPVIndicator .....	544
DPVIsCMRA .....	545
DPVIsNoStat .....	545
DPVIsVacant .....	546
DPVLocation .....	546
DPVResolveMultipleResponse .....	547
ErrorCode .....	547
ErrorCodeString .....	548
EWS .....	548

ExtraInfo .....	549
FirmOutput .....	549
GeocodeFootnote .....	550
HighwayContractFormat .....	551
IsResidence .....	551
KeepAliasAddress .....	552
KeepExtraPrimaryData .....	552
KeepNonMailingCity .....	553
LACS .....	554
LACSFootnote .....	554
LACSIndicator .....	555
Latitude .....	556
Longitude .....	556
LOTNumber .....	557
MailRoomServer .....	558
MatchedToDefault .....	558
MSACode .....	559
PlusCoded .....	559
PMB .....	560
PMBOutput .....	560
POBoxDeliveryOnlyZIP .....	561
POBoxFormat .....	561
PostDirectional .....	562
PostDirectionalFormat .....	562
PreDirectional .....	563
PreDirectionalFormat .....	563
PrimaryAddressOutput .....	564
PrimaryNumber .....	565
RecordType .....	565
ReturnAddressSuggestionList .....	566
ReturnInputAddressOnUnconfirmedDPV .....	567
RuralRouteFormat .....	567
SilentMode .....	568
State .....	568
Street .....	569
Suffix .....	569
SuffixFormat .....	570
SuiteLinkFootnote .....	570
UnitDesignator .....	571
UnitDesignatorFormat .....	571
UnitNumber .....	572
UnitOutput .....	573
UpdateUncorrectedCityStZip .....	574
Urbanization .....	574
UseDPV .....	575
UseGeocode .....	575
UseLACS .....	576
UseSuiteLink .....	577
ZipCode .....	577
ZipCoded .....	578
The .NET CASSAssembly Class for Correcting Batch Addresses .....	578
Using CASSAssembly .....	578
.NET CASSAssembly Functions .....	580
AbortTask .....	581

EndTask .....	581
GetProperty .....	582
GetPropertySummary .....	582
PrepareTask .....	583
PreviewCASSReports .....	583
PreviewReport .....	584
PrintCASSReports .....	585
PrintReport .....	585
RetrieveReviewed .....	586
ReviewErrors .....	587
SaveCASSReportsAsPDF .....	588
SaveReportAsPDF .....	589
SetProperty .....	590
ShowCassWizard .....	590
Update .....	591
ValidateProperties .....	592
.NET CASSAssembly Properties .....	592
CASS_ABBREVIATE_ADDRESS_LINE .....	593
CASS_CERTIFY_FLAG .....	593
CASS_DPV_FAILURE_AS_ERROR .....	594
CASS_DPV_RESOLVE_MULTIPLE_RESPONSE .....	594
CASS_DUAL_ADDRESS_INPUT_PREFERENCE .....	594
CASS_KEEP_ALIAS_ADDRESS .....	595
CASS_KEEP_EXTRA_PRIMARY_DATA .....	595
CASS_LIST_NAME .....	596
CASS_LIST_PROCESSOR .....	596
CASS_MAILERS_ADDRESS .....	596
CASS_MAILERS_CITY .....	596
CASS_MAILERS_NAME .....	597
CASS_MAILERS_STATE .....	597
CASS_MAILERS_ZIP .....	597
CASS_RETURN_INPUT_ADDRESS_ON_UNCONFIRMED_DPV .....	597
CASS_UPDATE_UNCORRECTED_CITY_ST_ZIP .....	598
CASS_USE_COUNTRY .....	598
CASS_USE_SUITELINK .....	598
FORMAT_CASING .....	599
FORMAT_CITY .....	599
FORMAT_FIRM_OUTPUT .....	600
FORMAT_HIGHWAY_CONTRACT .....	600
FORMAT_PMB_OUTPUT .....	601
FORMAT_PO_BOX .....	601
FORMAT_POST_DIRECTIONAL .....	602
FORMAT_PRE_DIRECTIONAL .....	602
FORMAT_PRIMARY_ADDRESS_OUTPUT .....	603
FORMAT_RURAL_ROUTE .....	603
FORMAT_SUFFIX .....	604
FORMAT_UNIT_DESIGNATOR .....	604
FORMAT_UNIT_OUTPUT .....	605
FORMAT_UPDATE_CASE_BUSINESS .....	605
FORMAT_UPDATE_CASE_NAMES .....	605
PRIMARY_ADDRESS_LENGTH_LIMIT .....	606
SETTINGS_CASS_WIZARD_CAPTION .....	606
SETTINGS_DATAFILE_LOCATION .....	606
SETTINGS_FIELD_LIST_IN .....	607

SETTINGS_FIELD_LIST_OUT .....	607
SETTINGS_HIDE_PROGRESS_AFTER_BATCH .....	607
SETTINGS_INI_FILE_NAME .....	608
SETTINGS_INPUT_BLOCK_RECORD_COUNT .....	608
SETTINGS_MAILROOM_SERVER_LIST .....	608
SETTINGS_MRTK_VERSION .....	609
SETTINGS_PRINT_ON_SERVER .....	609
SETTINGS_RECORD_COUNT .....	610
SETTINGS_REVIEW_ERRORS .....	610
SETTINGS_REVIEW_ERRORS_RECORD_COUNT_PER_RECEIVE .....	610
SETTINGS_REVIEW_ERRORS_SHOW_RECEIVE_PROGRESS .....	611
SETTINGS_SHOW_PROGRESS .....	611
SETTINGS_SILENT_MODE .....	611
.NET CASSAssembly Fields Summary Table .....	612
.NET CASSAssembly Reports Summary Table .....	621
The .NET MOVEAssembly Class for Updating Moved Addresses .....	622
Using .NET MOVEAssembly .....	622
.NET MOVEAssembly Functions .....	624
DoProcess .....	624
DoProcessEx .....	625
EndTask .....	625
GetProperty .....	626
PrepareTask .....	626
PreviewMoveReports .....	627
PreviewReport .....	627
PrintMoveReports .....	628
PrintReport .....	629
Retrieve .....	630
SaveMoveReportsAsPDF .....	631
SaveReportAsPDF .....	631
Send .....	632
SetProperty .....	633
ShowMoveWizard .....	634
ValidateProperties .....	635
.NET MOVEAssembly Properties .....	635
KEEP_ALTERNATE_ADDRESS_LINE .....	636
LOGIN_ADMIN_ID .....	636
LOGIN_ADMIN_PASSWORD .....	636
LOGIN_BROKER_ID .....	637
LOGIN_BROKER_PASSWORD .....	637
LOGIN_CUSTOMER_ID .....	637
LOGIN_CUSTOMER_PASSWORD .....	637
MOVE_BUYER_NAME .....	638
MOVE_CLIENT_ID_LIST .....	638
MOVE_CLIENT_ID_NAME_LIST .....	638
MOVE_HIGH_MATCH_RATE_REASON .....	639
MOVE_MAIL_CLASS .....	640
MOVE_MATCH_FLAG .....	641
MOVE_MOVE_MONTH_RANGE .....	641
MOVE_MULTI_NAME_HANDLE .....	642
NCOA_CUSTOMER_ADDRESS .....	642
NCOA_CUSTOMER_EMAIL .....	642
NCOA_CUSTOMER_FAX .....	642
NCOA_CUSTOMER_LASTLINE .....	643

NCOA_CUSTOMER_NAME .....	643
NCOA_CUSTOMER_PHONE .....	643
NCOA_CUSTOMER_SIC .....	643
NCOA_CUSTOMER_TITLE .....	643
SETTINGS_FIELD_LIST_IN .....	644
SETTINGS_FIELD_LIST_OUT .....	644
SETTINGS_HIDE_PROGRESS_AFTER_PROCESS .....	644
SETTINGS_INI_FILE_NAME .....	645
SETTINGS_INPUT_BLOCK_RECORD_COUNT .....	645
SETTINGS_MAILROOM_SERVER_LIST .....	645
SETTINGS_PRINT_ON_SERVER .....	646
SETTINGS_MRTK_VERSION .....	646
SETTINGS_SHOW_PROGRESS .....	646
.NET MOVEAssembly Reports .....	647
.NET MOVEAssembly Fields .....	647
The .NET PRESORTAssembly Class for Presorting Mailings .....	668
PRESORTAssembly Overview .....	668
PRESORTAssembly Functions .....	670
AbortTask .....	670
DoSort .....	670
EndTask .....	671
GetProperty .....	672
GetPropertySummary .....	672
PrepareTask .....	673
PreviewPresortReports .....	673
PreviewReport .....	675
PrintPresortReports .....	675
PrintReport .....	677
Retrieve .....	678
SavePresortReportsAsPDF .....	678
SaveReportAsPDF .....	680
Send .....	681
SetProperty .....	682
ShowPresortWizard .....	683
ValidateProperties .....	684
PRESORTAssembly Properties .....	685
CONTAINER_LABEL_MAILER_CITY_STATE .....	685
CONTAINER_LABEL_MAILER_COMPANY .....	685
CONTAINER_LABEL_MAILER_ZIP_CODE .....	685
CONTAINER_SACK_LABEL_COLUMNS .....	686
CONTAINER_SACK_LABEL_CONTINUOUS .....	686
CONTAINER_SACK_LABEL_HEIGHT .....	686
CONTAINER_SACK_LABEL_LEFT_MARGIN .....	687
CONTAINER_SACK_LABEL_ROWS .....	687
CONTAINER_SACK_LABEL_TOP_MARGIN .....	687
CONTAINER_SACK_LABEL_WIDTH .....	687
CONTAINER_TRAY_LABEL_COLUMNS .....	688
CONTAINER_TRAY_LABEL_CONTINUOUS .....	688
CONTAINER_TRAY_LABEL_HEIGHT .....	688
CONTAINER_TRAY_LABEL_LEFT_MARGIN .....	689
CONTAINER_TRAY_LABEL_ROWS .....	689
CONTAINER_TRAY_LABEL_TOP_MARGIN .....	689
CONTAINER_TRAY_LABEL_WIDTH .....	689
DATA_SERVICES_CLIENT_ID_LIST .....	690

DATA_SERVICES_PASSWORD .....	690
DATA_SERVICES_USER .....	691
PALLET_CREATE_LOW_VOLUME_PALLETS .....	691
PAYMENT_ACCOUNT_NUMBER .....	691
PAYMENT_OPTION .....	692
PERMIT_AGENT_ADDR .....	692
PERMIT_AGENT_CITY .....	693
PERMIT_AGENT_CONTACT .....	693
PERMIT_AGENT_EMAIL .....	693
PERMIT_AGENT_NAME .....	693
PERMIT_AGENT_PHONE .....	694
PERMIT_AGENT_STATE .....	694
PERMIT_AGENT_ZIP .....	694
PERMIT_HOLDER_ADDR .....	695
PERMIT_HOLDER_CAPS_CUSTOMER_ID .....	695
PERMIT_HOLDER_CITY .....	695
PERMIT_HOLDER_CONTACT .....	696
PERMIT_HOLDER_EMAIL .....	696
PERMIT_HOLDER_MAILER_ID .....	696
PERMIT_HOLDER_NAME .....	696
PERMIT_HOLDER_NONPROFIT_AUTH_NO .....	697
PERMIT_HOLDER_PHONE .....	697
PERMIT_HOLDER_STATE .....	697
PERMIT_HOLDER_ZIP .....	697
PERMIT_MAIL_OWNER_CRID .....	698
PERMIT_MAIL_OWNER_MAILER_ID .....	698
PERMIT_MAILING_AGENT_CRID .....	698
PERMIT_MAILING_AGENT_MAILER_ID .....	699
PERMIT_NUMBER .....	699
PERMIT_ORG_ADDR .....	699
PERMIT_ORG_CITY .....	700
PERMIT_ORG_CONTACT .....	700
PERMIT_ORG_EMAIL .....	700
PERMIT_ORG_MAILER_ID .....	701
PERMIT_ORG_NAME .....	701
PERMIT_ORG_NONPROFIT_AUTH_NO .....	701
PERMIT_ORG_PHONE .....	702
PERMIT_ORG_STATE .....	702
PERMIT_ORG_ZIP .....	702
PERMIT_OWNER_GHOST_NUMBER .....	702
PERMIT_POST_OFFICE .....	703
PERMIT_SECONDARY_CITY .....	703
PERMIT_SECONDARY_ACCOUNT_NUMBER .....	703
PERMIT_SECONDARY_NUMBER .....	704
PERMIT_SECONDARY_STATE .....	704
PERMIT_SECONDARY_ZIP .....	704
PRESORT_ACS_METHOD .....	705
PRESORT_ALWAYS_USE_PERIODICALS_FSS_PREP .....	706
PRESORT_ANNUAL_NONSUBSCRIBER_THRESHOLD_EXCEEDED .....	707
PRESORT_APPLY_PARCEL_SURCHARGE .....	707
PRESORT_APPLY_BREAK_MARK_INDICATOR .....	707
PRESORT_BREAK_MARK_IND_BUNDLE .....	708
PRESORT_BREAK_MARK_IND_CONTAINER .....	708
PRESORT_BREAK_MARK_IND_PALLET .....	708

PRESORT_CARTON_PIECE_MAX .....	709
PRESORT_CARTON_WEIGHT_MAX .....	709
PRESORT_CASS_PROCESS_DATE_AUTOMATION .....	709
PRESORT_CASS_PROCESS_DATE_ECR .....	709
PRESORT_COMBINE_RESIDUAL_PIECES .....	710
PRESORT_CONFIRM_TRACKING .....	710
PRESORT_CONTAINER_LABEL_LAYOUT .....	711
PRESORT_CREATE_COURTESY_PALLETS .....	711
PRESORT_CREATE_FSF_SACKS .....	711
PRESORT_CREATE_ORIGIN_DESTINATION_CONTAINERS .....	712
PRESORT_CUSTOM_BARCODE_MAILER_ID .....	712
PRESORT_DDU_ZIP .....	712
PRESORT_DROP_SHIP_BMC_ZIPS .....	713
PRESORT_DROP_SHIP_NDC_ZIPS .....	713
PRESORT_DROP_SHIP_SCF_ZIPS .....	713
PRESORT_EXCEPTIONAL_DISPATCH_ZIPS .....	714
PRESORT_FACING_SLIP_COLUMNS .....	714
PRESORT_FACING_SLIP_CONTINUOUS .....	714
PRESORT_FACING_SLIP_HEIGHT .....	715
PRESORT_FACING_SLIP_HORIZ_PITCH .....	715
PRESORT_FACING_SLIP_LEFT_MARGIN .....	715
PRESORT_FACING_SLIP_PAGE_HEIGHT .....	716
PRESORT_FACING_SLIP_PAGE_WIDTH .....	716
PRESORT_FACING_SLIP_PRINT_ORDER .....	716
PRESORT_FACING_SLIP_ROWS .....	716
PRESORT_FACING_SLIP_TOP_MARGIN .....	717
PRESORT_FACING_SLIP_VERT_PITCH .....	717
PRESORT_FACING_SLIP_WIDTH .....	717
PRESORT_FIRM_BUNDLE_MIN .....	717
PRESORT_FORCE_WALK_SEQUENCE_SATURATION .....	718
PRESORT_IM_BARCODE_MAILER_ID_CODE .....	718
PRESORT_IM_CONTAINER_SEQUENCE_LAST .....	719
PRESORT_IM_CONTAINER_SEQUENCE_START .....	719
PRESORT_IM_EINDUCTION .....	719
PRESORT_IM_EINDUCTION_ACCEPT_MISSHIPPED .....	720
PRESORT_IM_EINDUCTION_FAST_SCHEDULER_ID .....	720
PRESORT_IM_PALLET_SEQUENCE_LAST .....	721
PRESORT_IM_PALLET_SEQUENCE_START .....	721
PRESORT_IM_PIECE_SEQUENCE_LAST .....	721
PRESORT_IM_PIECE_SEQUENCE_START .....	722
PRESORT_IM_SEQUENCING_METHOD .....	722
PRESORT_IN_COUNTY_ZIPS .....	722
PRESORT_JOB_ID .....	723
PRESORT_LEGACY_DDU_SUPPORT .....	723
PRESORT_LIMITED_CIRCULATION .....	724
PRESORT_LIST_NAME .....	724
PRESORT_MAIL_CONTENT .....	724
PRESORT_MAILDAT_CONTACT_EMAIL .....	725
PRESORT_MAILDAT_CONTACT_NAME .....	725
PRESORT_MAILDAT_CONTACT_PHONE .....	725
PRESORT_MAILDAT_CREATE_PDR .....	726
PRESORT_MAILDAT_INFORMED_CODE .....	726
PRESORT_MAILDAT_INFORMED_END .....	726
PRESORT_MAILDAT_INFORMED_ID .....	727

PRESORT_MAILDAT_INFORMED_NAME .....	727
PRESORT_MAILDAT_INFORMED_REPRESENT .....	727
PRESORT_MAILDAT_INFORMED_RIDEALONG .....	728
PRESORT_MAILDAT_INFORMED_START .....	729
PRESORT_MAILDAT_INFORMED_TARGETURL .....	729
PRESORT_MAILDAT_INFORMED_TITLE .....	729
PRESORT_MAILDAT_MAILING_FACILITY_ID .....	729
PRESORT_MAILDAT_MAILING_TITLE .....	730
PRESORT_MAILDAT_PERMIT HOLDER_ID .....	730
PRESORT_MAILDAT_USER_LICENSE_CODE .....	730
PRESORT_MAILDAT_VERIFICATION_FACILITY_NAME .....	731
PRESORT_MAILDAT_VERIFICATION_FACILITY_ZIP4 .....	731
PRESORT_MAILDAT_VERSION .....	731
PRESORT_MAILER_ID_USED .....	732
PRESORT_MAILING_DATE .....	732
PRESORT_MAILING_DROP_ZIP_CODE .....	732
PRESORT_MAILING_JOB_ID .....	733
PRESORT_MAKE_ONLY_FULL_5_DIGIT_CARRIER_ROUTE_TRAYS .....	733
PRESORT_MANIFEST_SEQUENCE_NUMBER .....	733
PRESORT_MAX_PIECES_PER_BUNDLE .....	734
PRESORT_MIN_POSTAGE_AMOUNT .....	734
PRESORT_MIXED_WEIGHT_SORT .....	734
PRESORT_MOVE_UPDATE_DATE .....	735
PRESORT_MOVE_UPDATE_METHOD .....	735
PRESORT_MULTIPLE_ENTRY_POINT .....	736
PRESORT_PALLET_PLACARD_LAYOUT .....	736
PRESORT_PERMIT_ZIPCODE .....	737
PRESORT_PIECE_WEIGHT .....	737
PRESORT_PIECES_ARE_BARCODED .....	737
PRESORT_PIECES_PER_INCH .....	738
PRESORT_POSTALONE_JOB_ID .....	738
PRESORT_PREFERRED_CONTAINER_TYPE .....	738
PRESORT_PRINT_FACING_SLIPS .....	739
PRESORT_PRINT_PALLET_LABELS .....	739
PRESORT_PRINT_TRAY_LABELS_ONLY .....	740
PRESORT_PUB_INFO_CONTACT_NAME .....	740
PRESORT_PUB_INFO_CONTACT_PHONE .....	740
PRESORT_PUB_INFO_EDITION_ISSUE .....	741
PRESORT_PUB_INFO_ENTRY_STATE_ZIP4 .....	741
PRESORT_PUB_INFO_ISSUE_DATE .....	741
PRESORT_PUB_INFO_ISSUE_FREQUENCY .....	742
PRESORT_PUB_INFO_ISSUE_FREQUENCY_PER_YEAR .....	742
PRESORT_PUB_INFO_NON_ADVERTISING_PERCENT .....	742
PRESORT_PUB_INFO_PUB_NUM .....	742
PRESORT_PUB_INFO_RIDE_ALONG_WEIGHT .....	743
PRESORT_PUB_INFO_TITLE .....	743
PRESORT_PUB_INFO_VOLUME_NUMBER .....	743
PRESORT_PUB_OWNER_AGENT_NAME .....	744
PRESORT_REDUCE_OVERFLOW_CONTAINERS .....	744
PRESORT_SEQUENCING_DATE_ECR .....	744
PRESORT_SHIPPING_SERVICES_LOGIN_ID .....	745
PRESORT_SHIPPING_SERVICES_SEQUENCE_NUMBER .....	745
PRESORT_SIMPLIFIED_DELIVERY_STATS_DATE .....	745
PRESORT_SORT_RESULTS_FLAG .....	746

PRESORT_STATEMENT_NUMBER .....	746
PRESORT_SUPPRESS_CARRT_BASIC_RATES .....	747
PRESORT_SUPPRESS_IN_COUNTY_RATES .....	747
PRESORT_TARE_WEIGHT_PALLET .....	747
PRESORT_TARE_WEIGHT_PRIMARY_CONTAINER .....	748
PRESORT_TARE_WEIGHT_SECONDARY_CONTAINER .....	748
PRESORT_TEMPLATE_LIST .....	748
PRESORT_USE_ACS_NONE_FOR_SAMPLE_COPIES .....	749
PRESORT_USE_DETACHED_ADDRESS_LABEL .....	749
PRESORT_USE_EASYTRACK .....	750
PRESORT_USE_EMM_TRAYS .....	751
PRESORT_USE_EXCEPTIONAL_DISPATCH .....	751
PRESORT_USE_FIRM_BUNDLES .....	751
PRESORT_USE_IM_ONLY .....	752
PRESORT_USE_MAILDAT .....	752
PRESORT_USE_REPOSITIONABLE_NOTES .....	752
PRESORT_USE_SIMPLIFIED_ADDRESSING_PIECE_MAX .....	753
PRESORT_USE_USPS_PROMOTION .....	753
QUERY_BARCODE_REQUIRED .....	754
QUERY_ENDORSEMENT_REQUIRED .....	754
QUERY_IM_BARCODE_REQUIRED .....	755
REPORT_FILE_NAME_CONTAINER_LABELS .....	755
REPORT_FILE_NAME_MAILDAT .....	755
REPORT_FOLDER_NAME_MAILDAT .....	756
REPORT_MAILDAT_SAVE_AS_ZIP .....	756
REPORT_PRINT_ALL_REPORTS .....	756
REPORT_PRINT_CASS_FORM .....	757
REPORT_PRINT_CONTAINER_LABELS .....	757
REPORT_PRINT_CONTAINER_LISTING .....	757
REPORT_PRINT_DROP_SHIPMENT .....	758
REPORT_PRINT_MAIL_SORT_SUMMARY .....	758
REPORT_PRINT_MANIFEST .....	758
REPORT_PRINT_POSTAGE_STATEMENT .....	759
REPORT_PRINT_POSTAGE_SUMMARY .....	759
REPORT_PRINT_QUALIFICATION .....	760
REPORT_PRINT_ZIP_LISTING .....	760
REPORT_PRINT_ZONE .....	760
REPORT_PRINTER_CASS_FORM .....	761
REPORT_PRINTER_CONTAINER_LABELS .....	761
REPORT_PRINTER_CONTAINER_LISTING .....	761
REPORT_PRINTER_DEFAULT .....	761
REPORT_PRINTER_DROP_SHIPMENT .....	762
REPORT_PRINTER_FACING_SLIPS .....	762
REPORT_PRINTER_MAIL_SORT_SUMMARY .....	762
REPORT_PRINTER_MANIFEST .....	763
REPORT_PRINTER_PALLET_LABELS .....	763
REPORT_PRINTER_POSTAGE_STATEMENT .....	763
REPORT_PRINTER_POSTAGE_SUMMARY .....	763
REPORT_PRINTER_QUALIFICATION .....	764
REPORT_PRINTER_ZIP_LISTING .....	764
REPORT_PRINTER_ZONE .....	764
REPORT_SAVE_AS_FILE_CONTAINER_LABELS .....	765
REPORT_SAVE_MAILDAT .....	765
REPORT_SAVE_SHIPPING_SERVICES_FILE .....	765

REPORT_SHIPPING_SERVICES_FILE_PATH .....	766
SETTINGS_CASS_PROCESS_FIRST .....	766
SETTINGS_DATAFILE_LOCATION .....	766
SETTINGS_FIELD_LIST_IN .....	767
SETTINGS_FIELD_LIST_OUT .....	767
SETTINGS_HIDE_PROGRESS_AFTER_SORT .....	767
SETTINGS_INI_FILE_NAME .....	768
SETTINGS_INPUT_BLOCK_RECORD_COUNT .....	768
SETTINGS_MAILDAT_CREATE_PBC .....	768
SETTINGS_MAILROOM_SERVER_LIST .....	769
SETTINGS_MRTK_VERSION .....	769
SETTINGS_PAUSE_BEFORE_CONTAINER_LABELS .....	769
SETTINGS_PAUSE_BEFORE_FACING_SLIPS .....	770
SETTINGS_PAUSE_BEFORE_PALLET_LABELS .....	770
SETTINGS_PREFER_CONDENSED_REPORTS .....	770
SETTINGS_PRINT_ON_SERVER .....	771
SETTINGS_RECORD_COUNT .....	771
SETTINGS_RECORD_COUNT_PER_RECEIVE .....	771
SETTINGS_SHOW_CASS_FORM_CHECKBOX_IN_WIZARD .....	772
SETTINGS_SHOW_PRINT_DIALOG .....	772
SETTINGS_SHOW_SORT_PROGRESS .....	772
SETTINGS_SHOW_TEMPLATES_IN_WIZARD .....	773
SETTINGS_SILENT_MODE .....	773
SETTINGS_SORT_WIZARD_CAPTION .....	773
SETTINGS_TEMPLATE_NAME_TO_USE .....	773
PRESORTAssembly Reports Summary Table .....	774
PRESORTAssembly Fields .....	775

## The .NET ZIPAssembly Class for Correcting Single Addresses

The BCC Architect ZIPAssembly object verifies and corrects a single address. The ZIPAssembly object is designed for use with address entry forms, Web pages, and any other environment where single address verification is needed.

After checking an address, you can retrieve the corrected address information. This includes the delivery point, postal carrier route, and all the other individual elements of an address, if the address was corrected. Returning the individual elements is a powerful tool that can be used to test if certain elements are missing. For example, with just a few lines of code, you can tell if the user neglected to enter required unit information (e.g., suite or apartment number). If an address could not be verified, an error code and associated description can be retrieved, providing details about what caused the verification process to fail.

The ZIPAssembly object can act as a client of the BCC Architect Server software. This allows the matching process to be done on a server rather than the computer running the ZIPAssembly object. This is very useful in keeping the requirements low on the workstation. Putting the ZIPAssembly in client mode is as simple as setting the MailRoomServer property of the object. This property specifies where the BCC Architect Server is located and whether or not to use it. If you are using the

ZIPAssembly in a Web environment, it is highly recommended that you use the BCC Architect Server.

## Using ZIPAssembly for Single Address Correction

The following general procedure should be used to implement the ZIPAssembly class:

1. Add a reference to the .NET component Satori.MRTK.USAssembly.
2. While not required, you can reduce the amount of typing needed by adding using statements (C#) or Imports statements (Visual Basic) for the Satori.MailRoomToolKit.US and Satori.MailRoomToolKit.US Interfaces namespaces.
3. Create a ZIPAssembly object.
4. Call PrepareTask.
5. Set the properties for the input record. The minimum amount of data required is AddressLine1 plus either ZipCode, City and State, or CityStateZip. For best results, it is recommended that all of the basic address properties be used: BusinessName, AddressLine1, AddressLine2, City, State, and ZipCode (or CityStateZip). Alternatively, the entire address can be passed in at once using the AddressBlock property.
6. Set the formatting and configuration properties as desired. For example, you can set UseGeocode equal to true to return the latitude and longitude coordinates for an address or control the casing of the output address with the Casing property.
7. Call CheckAddress.
8. Retrieve the updated properties for the address. Of particular interest is the ErrorCode property, which provides information about the results of the address check.
9. Call EndTask.

## ZIPAssembly Functions

The ZipAssembly functions are defined below. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available ZIPAssembly members, and their definitions, in the Object Browser and IntelliSense.

## BrowseAddress

### Syntax

```
bool BrowseAddress();
```

### Description

Launches the Address Browser window. The Address Browser window is a tool that allows the user to search through all the addresses contained in the address files.

### Parameters

None.

### Return values

true

Indicates user clicked Update in the Address Browser window.

false

Indicates user clicked Cancel in the Address Browser window.

### Notes

This function will launch the Zip Browser window and try to "look up" the address contained within the ZIPAssembly object.

If you want to launch the Address Browser without automatically looking up an address then call `ClearAddress` before calling this function.

If the return value is true, then the ZIPAssembly object will contain the address elements for the address that the user chose to keep within the Address Browser window.

By checking the `ErrorCode` property after calling the `CheckAddress` function, the Address Browser window can be automatically launched for an address that does not correct.

### See also

- [CheckAddress](#)
- [ClearAddress](#)
- [ErrorCode](#)

## BuildAddress

### Syntax

```
void BuildAddress();
```

### Description

Builds an address from its individual address elements.

### Parameters

None.

### Return values

None.

### Notes

This function will build an address by combining the individual address elements, such as the primary number, street, suffix, etc., and update the AddressLine1 and AddressBlock properties.

The resulting address is not matched against the USPS database nor is any formatting applied to it.

### See also

- AddressBlock
- AddressLine1
- City
- CityStateZip
- PostDirectional
- PreDirectional
- PrimaryNumber
- State
- Street
- Suffix

- UnitDesignator
- UnitNumber
- ZipCode

## CheckAddress

### Syntax

```
void CheckAddress();
```

### Description

Correct and format an address.

### Parameters

None.

### Return values

None.

### Notes

This function will try to match the address contained in the ZIPAssembly object against the USPS database.

A call to CheckAddress will update all of the ZIPAssembly object property values.

After a call to CheckAddress, you can get the results of the process by checking the value of the ErrorCode property. In some situations, you may want to call BrowseAddress if the ErrorCode property indicates that the address could not be corrected.

The address elements returned after calling CheckAddress will be formatted according to the values of the various formatting properties.

### See also

- BrowseAddress
- ErrorCode

## ClearAddress

### Syntax

```
void ClearAddress();
```

### Description

Resets all of the ZIPAssembly object's property values.

### Parameters

None.

### Return values

None.

### Notes

This function can be used to quickly clear the address information from a ZIPAssembly object, thus bringing the object into a known state (i.e., no address, error code, etc.) before setting the object's properties.

### See also

- [CheckAddress](#)

## EndTask

### Syntax

```
void EndTask();
```

### Description

Cleans up and releases a ZIPAssembly object.

### Parameters

None.

### Return values

None.

### Notes

EndTask releases the address-matching engine.

It is recommended that you call `EndTask` when you are done with the `ZIPAssembly` object.

It is not required to call `EndTask` after each call to `CheckAddress`.

See also

## ParseAddress

Syntax

```
void ParseAddress();
```

Description

Parses an address into its individual address elements.

Parameters

None.

Return values

None.

Notes

This function will parse the input record and return the individual address elements, such as the primary number, street, suffix, etc., formatted according to the values of the various formatting properties.

The address is not matched against the USPS database.

See also

- [AddressBlock](#)
- [AddressLine1](#)
- [AddressLine2](#)
- [City](#)
- [CityFormat](#)
- [CityStateZip](#)
- [POBoxFormat](#)

- PostDirectional
- PostDirectionalFormat
- PreDirectional
- PreDirectionalFormat
- PrimaryNumber
- RuralRouteFormat
- State
- Street
- Suffixproperty
- SuffixFormat
- UnitDesignator
- UnitDesignatorFormat
- UnitNumber
- ZipCode

## PrepareTask

### Syntax

```
void PrepareTask();
```

### Description

Initializes and prepares the ZIPAssembly object.

### Parameters

None.

### Return values

None.

## Notes

PrepareTask should be called only once, after the ZIPAssembly object is created.

This function must be called before calling any of the other functions or setting any of the properties of the ZIPAssembly object. Failing to do so will cause subsequent function calls to fail. The one exception to this rule is the MailRoomServer property, which must be set prior to calling PrepareTask.

## See also

MailRoomServer

## .NET ZIPAssembly Properties

The ZipAssembly properties are defined below. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available ZIPAssembly members, and their definitions, in the Object Browser and IntelliSense.

### AddressBlock

#### Syntax

```
string AddressBlock;
```

#### Description

A read-write property containing the address as two or more lines of text.

## Notes

CheckAddress can be called with the address data input as an address block. This process will take two or more lines of text, identify which lines contain address data, check the address, and then reconstruct the address block using the formatted address lines, business name and last line (if the address was matched).

#### Input:

The address block must be a stream of delimited text. The end of each line in the address block should be delimited with a carriage return/line feed, line feed, carriage return or tab. The delimiter used must be consistent throughout the address block.

The following definitions are used to describe the lines in the block of text:

- Last Line – This line contains the city, state, and ZIP Code indicating what city the piece of mail should be delivered to.

- Address Line – This line is always required and specifies the actual delivery address. By the standards from USPS, this line should always be directly above the Last Line of the address.
- Unit Designator and Number – This line will only exist in the input record block and will only be recognized if the address line does not contain a unit designator and number.
- PMB – This line is information for a personal mailbox. This information is ignored during the certification process.

The algorithm for this process will first locate the address line by starting at the bottom-most line and then moving up until a recognizable ZIP Code or city/state combination is located. This line will be designated the last line of the address.

Once the last line has been located, the following schemes will be used to attempt to certify the address (listed in order of precedence). Items in brackets are optional. (For Puerto Rico addresses, the business name will be treated as the urbanization if it is recognizable as such.)

Scheme 1:

<Business Name>

<PMB>

<Unit Designator and Number>

Address Line

Last Line

Scheme 2:

<Business Name>

<PMB>

Address Line

<Unit Designator and Number>

Last Line

Scheme 3:

<Unit Designator and Number>

Address Line

Business Name or PMB

Last Line

Scheme 4:

Address Line

<Unit Designator and Number>

Business Name or PMB

Last Line

Output:

On output, the address block will be constructed according to the values of the various formatting properties. All spaces at the beginning and end of each line, and extra spaces between words will be removed. The output address block will be created with a carriage return/line feed as the delimiter.

Any lines of data that are not part of the corrected address will be returned in the address block. They will be placed in the reconstructed address block in the order that they were input and relative to (above/below) the address lines as they were input.

The following illustrates how the address block will be returned:

Unused Line 1

...

Unused Line N

<Business Name>

<Urbanization>

<PMB>

Address Line <Unit Designator and Number>

Last Line

Unused Line N+1

...

Unused Line M

#### See also

- [CheckAddress](#)
- [Casing](#)
- [KeepAliasAddress](#)
- [KeepExtraPrimaryData](#)
- [PMBOutput](#)
- [POBoxFormat](#)
- [PostDirectionalFormat](#)
- [PreDirectionalFormat](#)
- [PrimaryAddressOutput](#)
- [RuralRouteFormat](#)
- [SuffixFormat](#)
- [UnitDesignatorFormat](#)
- [UnitOutput](#)

## AddressInputPreference

### Syntax

```
int AddressInputPreference;
```

### Description

A read-write property that determines dual address preference.

### Notes

This property determines which address to give preference to if both a street address and a PO Box address are found and matched.

This property only comes into play if the two addresses are input as separate address lines. If they are input on the same address line, then USPS rules dictate that the PO Box address has priority, regardless of position.

This property should be set before calling `CheckAddress`.

The following table lists the available property values:

Value	Description
0	By position, giving preference to the bottom address when there are two address lines (default value)
1	Prefer PO Box address
2	Prefer street address

See also

- `CheckAddress`

## AddressLine1

Syntax

```
string AddressLine1;
```

Description

A read-write property containing the first address line.

Notes

`AddressLine1` is the top address line, above `AddressLine2`.

After calling `CheckAddress`, the value of the `PrimaryAddressOutput` property determines the location of the primary address line, if there is a single address line. If there are two address lines, `AddressLine2` will contain the primary address line and `AddressLine1` will contain the secondary address information.

Both `AddressLine1` and `AddressLine2` are formatted according to the values of the various formatting properties.

#### See also

- [CheckAddress](#)
- [AddressLine2](#)
- [Casing](#)
- [KeepAliasAddress](#)
- [KeepExtraPrimaryData](#)
- [PMBOutput](#)
- [POBoxFormat](#)
- [PostDirectionalFormat](#)
- [PreDirectionalFormat](#)
- [PrimaryAddressOutput](#)
- [RuralRouteFormat](#)
- [SuffixFormat](#)
- [UnitDesignatorFormat](#)
- [UnitOutput](#)

## AddressLine2

### Syntax

```
string AddressLine2;
```

### Description

A read-write property containing the second address line.

### Notes

AddressLine2 is the bottom address line, below AddressLine1 and above the city, state and ZIP Code.

After calling `CheckAddress`, the value of the `PrimaryAddressOutput` property determines the location of the primary address line, if there is a single address line. If there are two address lines, `AddressLine2` will contain the primary address line and `AddressLine1` will contain the secondary address information.

Both `AddressLine1` and `AddressLine2` are formatted according to the values of the various formatting properties.

See also

- `CheckAddress`
- `AddressLine1`
- `Casing`
- `KeepAliasAddress`
- `KeepExtraPrimaryData`
- `PMBOutput`
- `POBoxFormat`
- `PostDirectionalFormat`
- `PreDirectionalFormat`
- `PrimaryAddressOutput`
- `RuralRouteFormat`
- `SuffixFormat`
- `UnitDesignatorFormat`
- `UnitOutput`

## `AddressLineAbbreviated`

Syntax

```
bool AddressLineAbbreviated;
```

### Description

A read-write property that determines whether address lines return abbreviated or not.

### Notes

If set to true, then the primary address line will return abbreviated to no more than characters.

The default value for this property is false, no abbreviation.

### See also

- [CheckAddress](#)
- [AddressLine1](#)
- [AddressLine2](#)

## AddressSuggestionList

### Syntax

```
String AddressSuggestionList;
```

### Description

A delimited list of address suggestions that is returned when an input address is invalid and not able to be corrected.

### Notes

Call [CheckAddress](#) before attempting to retrieve the value of this property.

The field will display a maximum of 10 entries, all of which have had their deliverability confirmed (DPV).

The following table describes the delimiters that are used by default in the output text.

Delimiter Type	ASCII	Hex
Field	25	19
Record	26	1A

The type of input addresses that will generally return suggestions are those that have certain address elements that are missing or invalid, such as street suffixes and street directionals. Suggestions return the following fields:

- AddressLine 1
- City
- State
- ZIP Code

## ApplyCasingBusiness

### Syntax

```
bool ApplyCasingBusiness;
```

### Description

A read-write property that determines if the casing option is applied to the business name.

### Notes

You must call `CheckAddress` before the selected casing option is applied to the address elements in the `ZIPAssembly` object.

This property should be set before calling `CheckAddress`.

The default value for this property is `true`.

### See also

- `CheckAddress`
- `Casing`

## AssignLOT

### Syntax

```
bool AssignLOT;
```

### Description

A read-write property that determines if Line-of-Travel information should be returned.

## Notes

This property should be set if you want Line-of-Travel (LOT) information for an address. Doing so will slow down processing speed.

This property should be set before calling `CheckAddress`.

The default value for this property is `false`.

## See also

- `CheckAddress`
- `LOTNumber`

## AssignRDI

### Syntax

```
bool AssignRDI;
```

### Description

A read-write property that determines if the Residential Delivery Indicator should be returned.

## Notes

To use this property you must subscribe to the Residential Delivery Indicator (RDI) service through the USPS.

RDI is a BCC Architect add-on.

This property should be set before calling `CheckAddress`.

The default value for this property is `false`.

## See also

- `CheckAddress`
- `IsResidence`

## BusinessName

### Syntax

```
string BusinessName;
```

### Description

A read-write property containing the business name.

### Notes

The business name is used to match a firm record.

### See also

- [CheckAddress](#)
- [ApplyCasingBusiness](#)
- [Casing property](#)

## CarrierRoute

### Syntax

```
string CarrierRoute;
```

### Description

A read-write property containing the carrier route data for an address.

### Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

### See also

- [CheckAddress](#)

## CarrtCoded

### Syntax

```
bool CarrtCoded;
```

### Description

A read-only property that indicates if a carrier route was assigned to an address.

### Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)
- [CarrierRoute](#)

## Casing

Syntax

```
int casing;
```

Description

A read-write property that determines the casing format applied to the address elements.

Notes

You must call [CheckAddress](#) before the selected casing option is applied to the address elements in the [ZIPAssembly](#) object.

This property should be set before calling [CheckAddress](#).

The following table lists the available property values:

Value	Description
0	Upper case
1	Lower case
2	Mixed case (default value)

See also

- [CheckAddress](#)
- [ApplyCasingBusiness](#)

## CASSDate

Syntax

```
int CASSDate;
```

## Description

A read-only property containing information, such as the issue date, about the last attempt to check an address.

## Notes

If you are going to batch process it is strongly recommended that you save this data because it will allow a batch process (via CASSAssembly, for instance) to skip this record if it was previously coded with the current issue.

CheckAddress should be called before attempting to retrieve the value of this property.

## See also

- CheckAddress

## CensusBlock

### Syntax

```
string CensusBlock;
```

### Description

A read-only property containing the census block data for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The census block is returned when doing a geocode lookup. Geocode is a BCC Architect add-on.

### See also

- CheckAddress
- UseGeocode
- CensusTract
- GeocodeFootnote
- Latitude

- Longitude
- MSACode

## CensusTract

### Syntax

```
string CensusTract;
```

### Description

A read-only property containing the census tract data for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The census tract is returned when doing a geocode lookup. Geocode is a BCC Architect add-on.

### See also

- CheckAddress
- UseGeocode
- CensusBlock
- GeocodeFootnote
- Latitude
- Longitude
- MSACode

## City

### Syntax

```
string City;
```

### Description

A read-write property containing the city.

## Notes

CheckAddress should be called before attempting to retrieve the value of this property.

## See also

- CheckAddress
- CityFormat property

## CityFormat

### Syntax

```
int CityFormat;
```

### Description

A read-write property that determines the format of the city.

## Notes

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPAssembly object.

This property should be set before calling CheckAddress.

The following table lists the available property values:

Value	Description
0	Return full city name (default value)
1	Return abbreviated city name (if one exists)
2	Return as input THIS VALUE IS NO LONGER SUPPORTED

## See also

- CheckAddress
- AddressBlock

- [City](#)
- [CityStateZip](#)

## CityStateZip

### Syntax

```
string CityStateZip;
```

### Description

A read-write property containing the city, state and ZIP Code of an address.

### Notes

This property combines the individual city, state and ZIP Code elements that make up the last line of an address.

### See also

- [CheckAddress](#)
- [CityFormat](#)

## CongressionalDistrict

### Syntax

```
string CongressionalDistrict;
```

### Description

A read-only property containing the congressional district data for an address.

### Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

### See also

- [CheckAddress](#)

## CountyCode

### Syntax

```
int CountyCode;
```

### Description

A read-only property containing the county code for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

### See also

- CheckAddress
- CountyFIPSCode

## CountyFIPSCode

### Syntax

```
string CountyFIPSCode;
```

### Description

A read-only property containing the county FIPS code for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

This property contains the full 5-digit county FIPS code, where the first 2 digits represent the state and the last 3 digits represent the county.

### See also

- CheckAddress
- CountyCode

## CountyName

### Syntax

```
string CountyName;
```

### Description

A read-only property containing the county name for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

### See also

- CheckAddress

## DeliveryPointCheckDigit

### Syntax

```
string DeliveryPointCheckDigit;
```

### Description

A read-only property containing the delivery point and check digit for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

### See also

- CheckAddress

## BarcodeString

### Syntax

```
string DPBarcodeString;
```

### Description

A read-only property containing the delivery point barcode for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

Before a correct barcode can be printed or displayed on the screen, the SatBar.ttf font must be installed.

See also

- [CheckAddress](#)

## DPVCoded

Syntax

```
bool DPVCoded;
```

Description

A read-only property that indicates if the delivery point of an address was validated.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)
- [UseDPV](#)

## DPVFailureAsError

Syntax

```
bool DPVFailureAsError;
```

Description

A read-write property that determines how an address with unit information that does not confirm with DPV is handled.

Notes

Set to true to treat addresses that have a valid primary, but fail DPV because of missing or invalid unit information, as uncorrected addresses. The Zip+4 code will not be returned for these records.

This property should be set before calling [CheckAddress](#).

The default value for this property is false.

See also

- [CheckAddress](#)

## DPVFootnotes

### Syntax

```
string DPVFootnotes;
```

### Description

A read-only property that returns a code indicating specific information for a DPV lookup.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The following table lists the available property values:

Value	Description
LK	Processing locked out due to a seed record being processed
AA	Matched to the ZIP+4 file
A1	No match against the ZIP+4 file
BB	Matched to DPV file (all components confirmed)
CC	Matched only after removing secondary information, they were presented but invalid
N1	Input Primary matched, but high-rise missing secondary number
M1	Primary number missing
M3	Primary number invalid
P1	Input missing PO, RR, HC box number
P3	Failed DPV because of invalid PO, RR, or HC box number
RR	Matched CMRA (found in CMRA file)
R1	Matched CMRA, but secondary number (i.e., PMB) missing
U1	Matched unique ZIP Code
G1	Matched General delivery

Value	Description
F1	Matched military address

See also

- CheckAddress
- UseDPV

## DPVIndicator

Syntax

```
string DPVIndicator;
```

Description

A read-only property that returns a single character indicating the level of delivery point validity.

Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The following table lists the available property values:

Value	Description
Y	Both the primary and secondary (if present) validated against the DPV database.
S	The primary address is valid according to DPV, but the secondary is invalid.
D	The primary address is valid according to DPV, but the address is missing a secondary.
N	The primary address is not valid according to DPV.
""	The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not Zip+4 coded.
X	The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.
E	The DPV data file is more than 5 days old. By USPS restrictions, no more addresses can be presented to the DPV table.

See also

- [CheckAddress](#)
- [UseDPV](#)

## DPVIsCMRA

Syntax

```
bool DPVIsCMRA;
```

Description

A read-only property that indicates if the address is a commercial mail-receiving agent.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

The returned property value will always be false if the address is not DPV coded.

See also

- [CheckAddress](#)
- [UseDPV](#)

## DPVIsNoStat

Syntax

```
bool DPVIsNoStat;
```

Description

Indicates if the address is not receiving delivery and is not counted as a possible delivery. The address is not receiving delivery because: 1) delivery has not been established, 2) The customer receives mail as part of a drop, 3) the carrier destroys or returns all of the mail.

## Notes

### See also

- [CheckAddress](#)
- [UseDPV](#)

## DPVIsVacant

### Syntax

```
bool DPVIsVacant;
```

### Description

A read-only property that indicates if the address is unoccupied.

### Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

The returned property value will always be false if the address is not DPV coded.

### See also

- [CheckAddress](#)
- [UseDPV](#)

## DPVLocation

### Syntax

```
string DPVLocation;
```

### Description

A read-write property that contains the location of the DPV data file.

### Notes

The format of the returned property value is the full path name of the folder containing the DPV.cas file, ending with a backslash.

See also

## DPVResolveMultipleResponse

Syntax

```
bool DPVResolveMultipleResponse;
```

Description

THIS PROPERTY IS OBSOLETE. A read-write property that determines if DPV should be used to help resolve multiple responses.

Notes

Set to true to use DPV to help match addresses that would otherwise be uncoded because of multiple possibilities.

This property should be set before calling `CheckAddress`.

The default value for this property is false.

See also

- `CheckAddress`

## ErrorCode

Syntax

```
int ErrorCode;
```

Description

A read-only property containing a code that indicates the results of an address-matching attempt.

Notes

The error code indicates whether an address was matched to the USPS database. It also provides information about the changes, if any, that were made to the input record in order to correct it, or the reason an address was unable to be corrected.

An error code less than 0 indicates that the address was corrected, while an error code greater than 0 indicates that the address was unable to be corrected.

`CheckAddress` should be called before attempting to retrieve the value of this property.

See the Error Codes table for a complete list of error code values and their descriptions.

See also

- [CheckAddress](#)
- [ErrorCodeString](#)

## ErrorCodeString

Syntax

```
string ErrorCodeString;
```

Description

A read-only property containing a description of the results of an address-matching operation.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

The error description corresponds to the value of the [ErrorCode](#) property.

See also

- [CheckAddress](#)
- [ErrorCode](#)

## EWS

Syntax

```
bool EWS;
```

Description

A read-only property that indicates whether an address was flagged by the USPS Early Warning System (EWS).

Notes

The Early Warning System is used to prevent changes to an address that may be new construction but has not yet been added to the USPS database.

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)

## ExtraInfo

Syntax

```
string ExtraInfo;
```

Description

A read-only property containing the extra information from an input record.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)

## FirmOutput

Syntax

```
int FirmOutput;
```

Description

A read-write property that determines the location of the firm name.

Notes

This property determines how the business name is handled if it is not input as [BusinessName](#) and is recognized as a firm name.

This property should be set before calling [CheckAddress](#).

You must call [CheckAddress](#) before the selected formatting option is applied to the address elements in the [ZIPAssembly](#) object.

The following table lists the available property values:

Value	Description
0	Return as input (default value)

Value	Description
1	Return as BusinessName (only if input BusinessName was blank)
2	Always return as BusinessName (will overwrite input BusinessName)

See also

- CheckAddress
- BusinessName

### GeocodeFootnote

Syntax

```
string GeocodeFootnote;
```

Description

A read-only property containing additional information for a geocode lookup.

Notes

This property returns a code indicating the granularity of ZIP Code match (ZIP Code length) for a geocode lookup.

CheckAddress should be called before attempting to retrieve the value of this property.

Geocode is a BCC Architect add-on.

The following table lists the available property values:

Value	Description
00	Unable to perform a geocode lookup
03	Geocode data based on a 3-digit ZIP Code
05	Geocode data based on a 5-digit ZIP Code
07	Geocode data based on a 7-digit ZIP Code
09	Geocode data based on a 9-digit ZIP Code

## See also

- [CheckAddress](#)
- [UseGeocode](#) property

## HighwayContractFormat

## Syntax

```
int HighwayContractFormat;
```

## Description

A read-write property that determines the format of a highway contract address.

## Notes

This property should be set before calling [CheckAddress](#).

You must call [CheckAddress](#) before the selected formatting option is applied to the address elements in the [ZIPAssembly](#) object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## See also

- [CheckAddress](#)

## IsResidence

## Syntax

```
bool IsResidence;
```

### Description

A read-only property that indicates if the address is residential.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

To use this property you must subscribe to the Residential Delivery Indicator service through the USPS.

### See also

- CheckAddress
- AssignRDI

## KeepAliasAddress

### Syntax

```
bool KeepAliasAddress;
```

### Description

A read-write property that determines how a street name input as an alias should be returned.

### Notes

If property value is true, the input street name alias is returned. Otherwise, the "official" street name from the USPS database is returned.

This property should be set before calling CheckAddress.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPAssembly object.

The default value for this property is true.

### See also

- CheckAddress

## KeepExtraPrimaryData

### Syntax

```
bool KeepExtraPrimaryData;
```

### Description

A read-write property that determines if extra information in the primary address line is kept.

### Notes

This property should be set before calling `CheckAddress`.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

The default value for this property is `false`.

### See also

- `CheckAddress`

## KeepNonMailingCity

### Syntax

```
bool KeepNonMailingCity;
```

### Description

A read-write property that determines how a city input as a non-mailing city should be returned.

### Notes

If property value is `true`, a city name input as a non-mailing city (non-preferred) is returned. Otherwise, the USPS preferred city name is returned.

This property should be set before calling `CheckAddress`.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

### See also

- `CheckAddress`
- `City`
- `CityStateZip`

## LACS

### Syntax

```
bool LACS;
```

### Description

A read-only property that indicates if the address was flagged by the USPS Locatable Address Conversion System (LACS).

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

### See also

- CheckAddress

## LACSFootnote

### Syntax

```
string LACSFootnote;
```

### Description

A read-only property that contains a code indicating specific information for a LACS<sup>Link</sup> lookup.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The following table lists the available property values:

Value	Description
""	Not processed / Seed record
00	No match
09	Matched to default high-rise address; address not updated
14	Match failure to build new address
92	Match secondary dropped from input

Value	Description
A	Match success

See also

- CheckAddress
- UseLACS
- LACSIndicator

## LACSIndicator

Syntax

```
string LACSIndicator;
```

Description

A read-only property that contains a single character indicating the result of a LACS<sup>Link</sup> lookup.

Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The following table lists the available property values:

Value	Description
""	Not processed
N	No match / Matched, but there was a failure to build new address
Y	Match success
S	Match with secondary dropped from input
F	Seed record

#### See also

- [CheckAddress](#)
- [UseLACS](#)
- [LACSFootnote](#)

## Latitude

#### Syntax

```
string Latitude;
```

#### Description

A read-only property containing the latitude coordinate for an address.

#### Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

The latitude is returned when doing a geocode lookup.

Geocode is a BCC Architect add-on.

#### See also

- [CheckAddress](#)
- [UseGeocode](#)
- [CensusBlock](#)
- [CensusTract](#)
- [GeocodeFootnote](#)
- [Longitude](#)
- [MSACode](#)

## Longitude

#### Syntax

```
string Longitude;
```

## Description

A read-only property containing the longitude coordinate for an address.

## Notes

CheckAddress should be called before attempting to retrieve the value of this property.

The longitude is returned when doing a geocode lookup.

Geocode is a BCC Architect add-on.

## See also

- CheckAddress
- UseGeocode
- CensusBlock
- CensusTract
- GeocodeFootnote
- Latitude
- MSACode

## LOTNumber

### Syntax

```
string LOTNumber;
```

### Description

A read-only property containing the Line-Of-Travel (LOT) number for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

### See also

- CheckAddress

## MailRoomServer

### Syntax

```
string MailRoomServer;
```

### Description

A read-write property containing the location of the BCC Architect Server.

### Notes

Setting this property creates a TCP/IP connection to the BCC Architect Server, which can reside on the local network or virtually anywhere.

This property should be set before calling PrepareTask.

It is recommended that the BCC Architect Server be used when certifying addresses from a Web site.

The format is: Server Name (or IP Address):Port.

Currently, going outside of the proxy server might not be supported.

### See also

- PrepareTask

## MatchedToDefault

### Syntax

```
bool MatchedToDefault;
```

### Description

A read-only property that indicates if an address was matched to the default ZIP+4 code.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

This property is true if multiple responses exist and the address is matched to the default entry because of missing secondary information.

### See also

- CheckAddress

## MSACode

### Syntax

```
string MSACode;
```

### Description

A read-only property containing the metropolitan statistical area code for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

This MSACode is returned when doing a geocode lookup.

Geocode is a BCC Architect add-on.

### See also

- CheckAddress
- UseGeocode
- CensusBlock
- CensusTract
- GeocodeFootnote
- Latitude
- Longitude

## PlusCoded

### Syntax

```
bool PlusCoded;
```

### Description

A read-only property indicating if an address was corrected and assigned a ZIP+4 code.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)

## PMB

Syntax

```
string PMB;
```

Description

A read-only property containing the private mailbox number from an input record.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)

## PMBOutput

Syntax

```
int PMBOutput;
```

Description

A read-write property that determines the location of the private mailbox number.

Notes

This property should be set before calling [CheckAddress](#).

You must call [CheckAddress](#) before the selected formatting option is applied to the address elements in the [ZIPAssembly](#) object.

The following table lists the available property values:

Value	Description
0	Return on same line as unit information (default value)
1	Return as input

See also

- CheckAddress
- PMB

## POBoxDeliveryOnlyZIP

Syntax

```
int POBoxPOBoxDeliveryOnlyZIP;
```

Description

Indicates whether delivery is to an address in a PO Box only zone.

Notes

Returns 0 if False, -1 if True.

## POBoxFormat

Syntax

```
bool POBoxFormat;
```

Description

A read-write property that determines the format of a PO Box address.

Notes

This property should be set before calling CheckAddress.

You must call CheckAddress before the selected formatting option is applied to the address elements in the ZIPAssembly object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

See also

- [CheckAddress](#)

## PostDirectional

Syntax

```
string PostDirectional;
```

Description

A read-write property containing the postdirectional element.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

An example of a postdirectional would be the "E" in "Alaska Ave E."

See also

- [CheckAddress](#)

## PostDirectionalFormat

Syntax

```
int PostDirectionalFormat;
```

Description

Determines the format of the postdirectional element of an address.

Notes

This property should be set before calling [CheckAddress](#).

You must call [CheckAddress](#) before the selected formatting option is applied to the address elements in the [ZIPAssembly](#) object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)

Value	Description
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

See also

- [CheckAddress](#)
- [PostDirectional](#)

## PreDirectional

Syntax

```
string PreDirectional;
```

Description

A read-write property containing the predirectional element.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

An example of a predirectional would be the "W" in "W 6th Street."

See also

- [CheckAddress](#)

## PreDirectionalFormat

Syntax

```
int PreDirectionalFormat;
```

Description

A read-write property that determines the format of the predirectional element of an address.

Notes

This property should be set before calling [CheckAddress](#).

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

See also

- `CheckAddress`
- `PreDirectional`

## PrimaryAddressOutput

Syntax

```
int PrimaryAddressOutput;
```

Description

A read-write property that determines the location of the primary address.

Notes

This property should be set before calling `CheckAddress`.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

The following table lists the available property values:

Value	Description
0	Return as <code>AddressLine1</code> (if <code>AddressLine2</code> is blank; otherwise, the primary address line will be returned as <code>AddressLine2</code> with the secondary address data in <code>AddressLine1</code> ); this is the default value

Value	Description
1	Return as AddressLine2

#### See also

- CheckAddress
- AddressLine1
- AddressLine2

## PrimaryNumber

### Syntax

```
string PrimaryNumber;
```

### Description

A read-write property containing the primary number.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

An example of a primary number would be the "" in " 2nd Ave."

### See also

- CheckAddress

## RecordType

### Syntax

```
string RecordType;
```

### Description

A read-only property that indicates the record type for an address.

### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

This property returns a USPS flag that represents the type of address that is contained in the ZIPAssembly object. For example, a PO BOX address will return a value of "P."

The following table lists the available property values:

Value	Description
S	Street record
P	Post Office box
R	Rural Route or Highway Contract
H	High-rise, Building, or Apartment
F	Firm Record
G	General Delivery
M	Multi-Carrier Record

See also

- [CheckAddress](#)

## ReturnAddressSuggestionList

Syntax

```
bool ReturnAddressSuggestionList;
```

Description

Indicates whether to return a list of suggested addresses for an input address that is invalid and can't be corrected.

Notes

You can read and set this property.

This property should be set after calling `PrepareTask`, but before `ValidateProperties`.

This property should be set to `True` if you want to turn on the `AddressSuggestionList` property, which provides a list of suggested addresses returned for an input address that is invalid.

Default setting for this property is `False`.

See also

- [CheckAddress](#)

## ReturnInputAddressOnUnconfirmedDPV

Syntax

```
bool ReturnInputAddressOnUnconfirmedDPV;
```

Description

Determines whether to roll back corrected addresses that are not DPV confirmable.

Notes

This property should be set before calling [CheckAddress](#).

Default setting for this property is `False`. When set to `True`, addresses that cannot be verified for purposes of DPV are returned unchanged with casing applied, that is, address correction rolls back to the original address. The error code indicates why the address is not verified.

In addition, when set to `True`, the setting of the `DPVFailureAsError` is ignored. However, if the original address includes a ZIP+4, the returned output will not include the +4 part of the ZIP Code.

See also

- [CheckAddress](#)

## RuralRouteFormat

Syntax

```
int RuralRouteFormat;
```

Description

A read-write property that determines the format of a rural route address.

Notes

This property should be set before calling [CheckAddress](#).

You must call [CheckAddress](#) before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

See also

- [CheckAddress](#)

## SilentMode

Syntax

```
bool SilentMode;
```

Description

A read-write property that determines ZIPAssembly's mode of operation.

Notes

Running ZIPAssembly in silent mode disables all dialogs, including error messages.

This property should be set before calling [CheckAddress](#).

The default value for this property is false.

See also

## State

Syntax

```
string State;
```

Description

A read-write property containing the state.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

See also

- [CheckAddress](#)

## Street

Syntax

```
string Street;
```

Description

A read-write property containing the street.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

An example of a street would be "Lake" in "4 Lake St."

See also

- [CheckAddress](#)

## Suffix

Syntax

```
string Suffix;
```

Description

A read-write property containing the suffix.

Notes

[CheckAddress](#) should be called before attempting to retrieve the value of this property.

An example of a suffix would be "St" in "4 Lake St."

See also

- [CheckAddress](#)
- [SuffixFormat](#)

## SuffixFormat

### Syntax

```
int SuffixFormat;
```

### Description

A read-write property that determines the format of the suffix element of an address.

### Notes

This property should be set before calling `CheckAddress`.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

### See also

- [CheckAddress](#)
- [Suffix](#)

## SuiteLinkFootnote

### Syntax

```
string SuiteLinkFootnote;
```

### Description

A read-only property that contains a code indicating specific information for a Suite<sup>Link</sup> lookup.

### Notes

`CheckAddress` should be called before attempting to retrieve the value of this property.

The following table lists the available property values:

Value	Description
""	Was not processed by the Suite <sup>Link</sup> engine: the address did not qualify for a lookup within the Suite <sup>Link</sup> file. Only default high-rise addresses qualify for a Suite <sup>Link</sup> lookup.
A	The address was processed and secondary information was added to the resulting address.
	The address was processed through the Suite <sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.

See also

- [CheckAddress](#)
- [UseSuiteLink](#)

## UnitDesignator

Syntax

```
string UnitDesignator;
```

Description

A read-write property containing the unit designator.

Notes

CheckAddress should be called before attempting to retrieve the value of this property.

An example of a unit designator would be "Ste" in " 2nd Ave Ste 0."

See also

- [CheckAddress](#)

## UnitDesignatorFormat

Syntax

```
int UnitDesignatorFormat;
```

### Description

A read-write property that determines the format of the unit designator.

### Notes

This property should be set before calling `CheckAddress`.

You must call `CheckAddress` before the selected formatting option is applied to the address elements in the `ZIPAssembly` object.

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

### See also

- `CheckAddress`
- `UnitDesignator` property

## UnitNumber

### Syntax

```
string UnitNumber;
```

### Description

A read-write property containing the unit number.

### Notes

`CheckAddress` should be called before attempting to retrieve the value of this property.

An example of a unit number would be "0" in " 2nd Ave Ste 0."

See also

- [CheckAddress](#)

## UnitOutput

Syntax

```
int UnitOutput;
```

Description

A read-write property that determines the location of the unit information.

Notes

This property should be set before calling [CheckAddress](#).

You must call [CheckAddress](#) before the selected formatting option is applied to the address elements in the [ZIPAssembly](#) object.

The following table lists the available property values:

Value	Description
0	Return at end of primary address line (default value)
1	Return on secondary address line (if secondary address is blank)
2	Always return on secondary address line (existing secondary address data will be overwritten)

See also

- [CheckAddress](#)
- [AddressLine1](#)
- [AddressLine2](#)
- [UnitDesignator](#)
- [UnitNumber](#)

## UpdateUncorrectedCityStZip

### Syntax

```
bool UpdateUncorrectedCityStZip;
```

### Description

A read-write property that determines if the city, state and ZIP Code data should be updated for an uncorrected address.

### Notes

If property value is true, the city, state and ZIP Code data will be returned, when possible, for addresses that are not corrected when `CheckAddress` is called.

This property should be set before calling `CheckAddress`.

The default value for this property is false.

### See also

- `CheckAddress`
- `City`
- `CityStateZip`
- `State`
- `ZipCode`

## Urbanization

### Syntax

```
string Urbanization;
```

### Description

A read-write property containing the urbanization data for a Puerto Rico address.

### Notes

`CheckAddress` should be called before attempting to retrieve the value of this property.

Urbanization data is used in Puerto Rico addresses only.

See also

- [CheckAddress](#)

## UseDPV

Syntax

```
bool UseDPV;
```

Description

A read-write property that determines if an address should be validated with DPV.

Notes

If property value is true, the address is checked for Delivery Point Validation (DPV) when [CheckAddress](#) is called.

This property should be set before calling [CheckAddress](#).

The default value for this property is true, as DPV is now an integral and essential component of address correction as required by the USPS.

See also

- [CheckAddress](#)
- [DPVCoded](#)
- [DPVFailureAsError](#)
- [DPVFootnotes](#)
- [DPVIndicator](#)
- [DPVIsCMRA](#)

## UseGeocode

Syntax

```
bool UseGeocode;
```

Description

A read-write property that determines if a geocode lookup should be performed for an address.

## Notes

If property value is true, geocode data will be returned when CheckAddress is called.

This property should be set before calling CheckAddress.

The default value for this property is false.

Geocode is a BCC Architect add-on.

## See also

- CheckAddress
- CensusBlock
- CensusTract
- GeocodeFootnote
- Latitude
- Longitude
- MSACode

## UseLACS

### Syntax

```
bool UseLACS;
```

### Description

A read-write property that determines if a LACS<sup>Link</sup> lookup should be performed for an address.

## Notes

If property value is true, LACS<sup>Link</sup> data will be returned when CheckAddress is called.

This property should be set before calling CheckAddress.

The default value for this property is true.

#### See also

- [CheckAddress](#)
- [LACSFootnote](#)
- [LACSIndicator](#)

## UseSuiteLink

#### Syntax

```
bool UseSuiteLink;
```

#### Description

A read-write property that determines if a Suite<sup>Link</sup> lookup should be performed for an address.

#### Notes

If property value is true, Suite<sup>Link</sup> will attempt to add missing secondary information when CheckAddress is called.

This property should be set before calling CheckAddress.

The default value for this property is True in batch processing, and is False in single implementations.

#### See also

- [CheckAddress](#)
- [SuiteLinkFootnote](#)

## ZipCode

#### Syntax

```
string ZipCode;
```

#### Description

A read-write property containing the Zip Code.

#### Notes

CheckAddress should be called before attempting to retrieve the value of this property.

After a successful call to `CheckAddress`, the `ZipCode` property includes the +4 code.

See also

- `CheckAddress`

## ZipCoded

Syntax

```
bool ZipCoded;
```

Description

A read-only property that indicates an address was coded with only a 5-digit ZIP Code.

Notes

`CheckAddress` should be called before attempting to retrieve the value of this property.

See also

- `CheckAddress`
- `PlusCoded`
- `ZipCode`

## The .NET CASSAssembly Class for Correcting Batch Addresses

The BCC Architect `CASSAssembly` object performs address verification and correction on a batch of addresses. `CASSAssembly` provides a flexible interface that allows you to control the amount of information returned for each address as well as the number of records processed at a time. `CASSAssembly` can be configured to skip records that were processed successfully on a previous occasion with the current issue. By doing so, you can avoid a costly rewrite to a database, thereby increasing processing speed.

### Using `CASSAssembly`

The following general procedure should be used to implement the `CASSAssembly` class:

1. Add a reference to the .NET component Satori.MRTK.USAssembly.
2. While not required, you can reduce the amount of typing needed by adding using statements (C#) or Imports statements (Visual Basic) for the Satori.MailRoomToolKit.US and Satori.MailRoomToolKit.US Interfaces namespaces.
3. Create a CASSAssembly object.
4. Call PrepareTask.
5. Create a USAddressFieldList object that defines the input fields for each record. Fields are added to the USAddressFieldList object using the USFields.Field enumeration. The minimum amount of data required is ADDRESS\_LINE\_1 plus either ZIP\_CODE, CITY and STATE, or LAST\_LINE. For best results, it is recommended that all of the basic address fields be used: BUSINESS, ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE, and ZIP\_CODE (or LAST\_LINE). Alternatively, the entire address can be passed in at once with the ADDRESS\_BLOCK field.
6. Create a USAddressFieldList object that defines the updated output fields for each processed record. Typically, this would include the basic address fields plus any of the data fields that result from the address matching process that are of use to you (e.g., CARRIER\_ROUTE or COUNTY\_NAME). Also of particular interest is the ERROR\_CODE field, which provides information about the results of the address check.
7. Call SetProperty to set the desired properties. The individual properties are specified using the USProperties.CASS enumeration. You must set SETTINGS\_FIELD\_LIST\_IN and SETTINGS\_FIELD\_LIST\_OUT using the USAddressFieldList objects that were created in steps 5 and 6, respectively. You should also specify the number of records to be processed at a time with SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT.
8. Call ValidateProperties.
9. Create a USAddressRecordBlock object.
10. Loop through the records in your database, and for each record to be processed:
11. Create a USAddressRecord object. A USAddressFieldList object is required when creating a USAddressRecord object. Use the USAddressFieldList object created in step 5.
12. Set the input field values using the Fields property of the USAddressRecord object.

13. Add the USAddressRecord object to the USAddressRecordBlock object.
14. Repeat steps a through c until the USAddressRecordBlock object contains SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT USAddressRecord objects.
15. Call Update, passing in the USAddressRecordBlock object. A new, updated USAddressRecordBlock object will be returned.
16. Iterate through the USAddressRecordBlock object to get each updated USAddressRecord object. In turn, iterate through each USAddressRecord object to get the updated fields for each record. The fields for each output record will match those specified by the SETTINGS\_FIELD\_LIST\_OUT property set earlier.
17. Repeat steps through until all records have been processed. Call the Clear function of the USAddressRecordBlock object to reset it before adding a new block of records to it.
18. If you have elected to review uncoded records (as determined by the SETTINGS\_REVIEW\_ERRORS property):
19. Call ReviewErrors to show the review window.
20. Call RetrieveReviewed. A new, updated USAddressRecordBlock object containing SETTINGS\_REVIEW\_ERRORS\_RECORD\_COUNT\_PER\_RECEIVE records will be returned.
21. Iterate through the USAddressRecordBlock object to get each updated USAddressRecord object. In turn, iterate through each USAddressRecord object to get the updated fields for each record. The fields for each output record will match those specified by the SETTINGS\_FIELD\_LIST\_OUT property set earlier.
22. Repeat steps b and c until all reviewed records have been retrieved.
23. Call PreviewCASSReports or PrintCASSReports to preview or print the CASS Summary Report (PS Form ). You can also save the report as a PDF file using the SaveCASSReportsAsPDF function.
24. Call EndTask.

## .NET CASSAssembly Functions

The CASSAssembly functions are defined below. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available CASSAssembly members, and their definitions, in the Object Browser and IntelliSense.

## AbortTask

### Syntax

```
void AbortTask();
```

### Description

Aborts processing.

### Parameters

None.

### Return values

None.

### Notes

Call this function to end processing prematurely.

### See also

## EndTask

### Syntax

```
void EndTask();
```

### Description

Cleans up and releases a CASSAssembly object.

### Parameters

None.

### Return values

None.

### Notes

EndTask cleans up and releases used resources and should be called when done with the CASSAssembly object.

You must call EndTask after processing the last batch of addresses.

See also

## GetProperty

Syntax

```
string GetProperty(USProperties.CASS cassPropertyId);
```

Description

Retrieves the current value of a CASSAssembly property.

Parameters

*cassPropertyId*

Specifies the enum name of the property to get.

Return values

The value of the property specified by *cassPropertyId*.

Notes

All property values are returned as a string, regardless of the data type passed into SetProperty.

See also

See the CASSAssembly Properties section for a complete list of properties.

## GetPropertySummary

Syntax

```
string GetPropertySummary(USProperties.CASS cassPropertyId);
```

Description

Returns a string containing a description of a property.

Parameters

*cassPropertyId*

Specifies the enum name of the property to get information for.

Return values

A description of the property specified by *cassPropertyId*.

## Notes

This function can be used to display the description of a property on screen to the user.

## See also

See the [CASSAssembly Properties](#) section for a complete list of properties.

## PrepareTask

### Syntax

```
void PrepareTask();
```

### Description

Initializes and prepares the CASSAssembly object.

### Parameters

None.

### Return values

None.

## Notes

PrepareTask should be called only once, after the CASSAssembly object is created.

This function must be called before calling any of the other functions or setting any of the properties of CASSAssembly. Failing to do so will cause subsequent function calls to fail. The only exceptions to this rule are setting the `SETTINGS_DATAFILE_LOCATION` and `SETTINGS_MAILROOM_SERVER_LIST` properties, which must be defined prior to calling PrepareTask.

## See also

## PreviewCASSReports

### Syntax

```
void PreviewCASSReports();
```

### Description

Previews all CASS reports.

### Parameters

None.

#### Return values

None.

#### Notes

The CASS Summary Report (PS Form ) is the only CASS report currently available.

#### See also

[PreviewReport](#)

### PreviewReport

#### Syntax

```
void PreviewReport(USReports.CASS mrtkReportId);
```

#### Description

Previews a CASS report.

#### Parameters

*mrtkReportId*

The ID of the report to preview.

#### Return values

None.

#### Notes

Unlike [PreviewCASSReports](#), which previews all of the CASS reports, this function only displays the specified report.

The CASS Summary Report (PS Form ) is the only CASS report currently available.

#### See also

- [PreviewCASSReports](#)
- See the [CASSAssembly Reports](#) section for a complete list of reports.

## PrintCASSReports

### Syntax

```
void PrintCASSReports(string printerName, bool showPrintSetupDialog);
```

### Description

Prints all CASS reports.

### Parameters

*printerName*

The name of the printer you wish to print to.

*showPrintSetupDialog*

Set to true to show the printer setup dialog or false to hide this dialog.

### Return values

None.

### Notes

If *printerName* is set to an empty string then the default printer will be used to print the form.

If *showPrintSetupDialog* is set to true then the value for *printerName* will be ignored.

If the printer is a network printer, the *printerName* parameter must be set to the full name (e.g., \\Server\HP Laser Jet 5). To find the full printer name, you could print a test page from the print driver.

The CASS Summary Report (PS Form ) is the only CASS report currently available.

### See also

PrintReport

## PrintReport

### Syntax

```
void PrintReport(USReports.CASS mrtkReportId, string printerName, bool showPrintSetupDialog);
```

### Description

Prints a CASS report.

## Parameters

*mrtkReportId*

The ID of the report to print.

*printerName*

The name of the printer you wish to print to.

*showPrintSetupDialog*

Set to true to show the printer setup dialog or false to hide this dialog.

## Return values

None.

## Notes

If *printerName* is set to an empty string then the default printer will be used to print the form.

If *showPrintSetupDialog* is set to true then the value for *printerName* will be ignored.

If the printer is a network printer, the *printerName* parameter must be set to the full name (e.g., \\Server\HP Laser Jet 5). To find the full printer name, you could print a test page from the print driver.

Unlike `PrintCASSReports`, which prints all of the CASS reports, this function only prints the specified report.

The CASS Summary Report (PS Form ) is the only CASS report currently available.

## See also

- `PrintCASSReports`
- See the `CASSAssembly Reports` section for a complete list of reports.

## RetrieveReviewed

### Syntax

```
USAddressRecordBlock RetrieveReviewed();
```

### Description

Retrieves the addresses kept by the user in the Review Errors window.

### Parameters

None.

### Return values

A USAddressRecordBlock object containing the updated addresses.

### Notes

Only those addresses the user chooses to keep in the Review Errors window will be returned by this function.

The number of addresses in the returned block is determined by the value of SETTINGS\_REVIEW\_ERRORS\_RECORD\_COUNT\_PER\_RECEIVE.

### See also

- ReviewErrors
- See the CASSAssembly Properties section for the definition of:
  - SETTINGS\_REVIEW\_ERRORS\_RECORD\_COUNT\_PER\_RECEIVE

## ReviewErrors

### Syntax

```
void ReviewErrors();
```

### Description

Displays the Review Errors window.

### Parameters

None.

### Return values

None.

### Notes

When the Review Errors window is displayed, any addresses that are kept will be returned by the RetrieveReviewed function.

The SETTINGS\_REVIEW\_ERRORS property should be set to true.

See also

[RetrieveReviewed](#)

## [SaveCASSReportsAsPDF](#)

Syntax

```
string SaveCASSReportsAsPDF(string fileName, int fileOption);
```

Description

Saves all CASS reports as a PDF file.

Parameters

*filename*

The name, including path, of the file to save.

*fileOption*

Specifies what to do if the supplied file name already exists:

- 0 – Overwrite if file with file name specified in fileName already exists.
- 1 – Prompt if file with file name specified in fileName already exists.
- 2 – Create new file if file with name specified in fileName already exists; a number will be appended to the file name.
- 3 – Append if file with name specified in fileName already exists (not yet implemented)

Return values

Returns a string containing the path and name of the saved file.

Notes

The CASS Summary Report (PS Form ) is the only CASS report currently available.

See also

[SaveReportAsPDF](#)

## SaveReportAsPDF

### Syntax

```
string SaveReportAsPDF(USReports.CASS mrtkReportId, string fileName,  
int fileOption);
```

### Description

Saves a report as a PDF file.

### Parameters

*mrtkReportId*

The report to save, as specified by the USReports.CASS enum.

*filename*

The name, including path, of the file to save.

*fileOption*

Specifies what to do if the supplied file name already exists:

- 0 – Overwrite if file with file name specified in fileName already exists.
- 1 – Prompt if file with file name specified in fileName already exists.
- 2 – Create new file if file with name specified in fileName already exists; a number will be appended to the file name.
- 3 – Append if file with name specified in fileName already exists (not yet implemented)

### Return values

Returns a string containing the path and name of the saved file.

### Notes

Unlike SaveCASSReportsAsPDF, which saves all of the CASS reports, this function only saves the specified report.

The CASS Summary Report (PS Form ) is the only CASS report currently available.

### See also

SaveCASSReportsAsPDF

See the CASSAssembly Reports section for a complete list of reports.

## SetProperty

### Syntax

```
void SetProperty(USProperties.CASS cassPropertyId,  
object val)
```

### Description

Sets the value of a CASSAssembly property.

### Parameters

*cassPropertyId*

Specifies the enum name of the property to set.

*val*

Specifies the new property value.

### Return values

None.

### Notes

The parameter *val* accepts multiple data types. See the CASSAssembly Properties section for the data type of each property.

### See also

See the CASSAssembly Properties section for a complete list of properties.

## ShowCassWizard

### Syntax

```
void ShowCassWizard();
```

### Description

Displays the CASS Wizard dialog.

### Parameters

None.

### Return values

None.

### Notes

Call this function if you want to display the CASS Wizard dialog. The CASS Wizard provides a graphical interface that leads a user through the various processing options.

### See also

## Update

### Syntax

```
USAddressRecordBlock Update(USAddressRecordBlock addressBlock);
```

### Description

Corrects the addresses contained in the address block.

### Parameters

*addressBlock*

A USAddressRecordBlock object containing the input records to be processed.

### Return values

A USAddressRecordBlock object containing the corrected addresses.

### Notes

You may want to experiment with the SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT property. In preliminary tests, we have found the optimal setting to be around 25 - 50. The number of fields that you want returned (see SETTINGS\_FIELD\_LIST\_OUT property) greatly affects this number.

For optimal performance, only ask for the output fields that you need. The reason for this is that the extra information requires additional lookups that slow processing.

### See also

See the CASSAssembly Properties section for the definition of:

- SETTINGS\_FIELD\_LIST\_IN
- SETTINGS\_FIELD\_LIST\_OUT

- SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT
- See the Field Names table for the definition of:
- SKIPPED\_CERTIFY

## ValidateProperties

### Syntax

```
void ValidateProperties();
```

### Description

Verifies that the CASSAssembly object is set up correctly and ready to run.

### Parameters

None.

### Return values

None.

### Notes

This function verifies that the basic requirements of a CASSAssembly object have been met:

The address matching engine is loaded and able to run.

The input field list consists of the minimum set of fields, namely, ADDRESS\_LINE\_1 and either CITY/STATE or ZIP\_CODE or LAST\_LINE.

This function needs to be called before you call Update.

### See also

SetProperty function

See the CASSAssembly Properties section for a complete list of properties.

## .NET CASSAssembly Properties

The CASSAssembly properties are members of the USProperties.CASS enumeration and are defined below. These enum names are used as arguments of the SetProperty and SetProperty functions. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USProperties.CASS enums in the Object Browser and IntelliSense.

## CASS\_ABBREVIATE\_ADDRESS\_LINE

### Data type

bool

### Description

Determines if the primary address line will be return abbreviated to no more than characters.

### Notes

If this property is true, the primary address line will be return abbreviated to no more than characters. Address lines less than characters will not be abbreviated.

The default value for this property is false, which means address lines will not be abbreviated.

This property affects the primary address line as well as the address block fields.

## CASS\_CERTIFY\_FLAG

### Data type

int

### Description

Determines if previously corrected records should be skipped in order to increase processing speed.

### Notes

Setting this property to 0 speeds up processing of a batch of addresses by skipping those addresses that have already been corrected with the current issue.

The CASSDATE field must be supplied as input in order for an address to be skipped.

The following table lists the available property values:

Value	Description
0	Check only those records not corrected with this issue (default value)
1	Check every record
4	Rebuild PS Form based on the CASSDATE field. All records passed in will be checked to see if they qualify to be added to the ZIP+4 Coded total.

## CASS\_DPV\_FAILURE\_AS\_ERROR

### Data type

bool

### Description

Determines how an address with unit information that does not confirm with DPV is handled.

### Notes

Set to true to treat addresses that have a valid primary, but fail DPV because of missing or invalid unit information, as uncorrected addresses. The Zip+4 code will not be returned for these records.

The default value for this property is false.

## CASS\_DPV\_RESOLVE\_MULTIPLE\_RESPONSE

### Data type

bool

### Description

THIS PROPERTY IS OBSOLETE. Determines if DPV should be used to help resolve multiple responses when address matching.

### Notes

Set to true to use DPV to help match addresses that would otherwise be uncoded because of multiple possibilities.

The default value for this property is false.

## CASS\_DUAL\_ADDRESS\_INPUT\_PREFERENCE

### Data type

int

### Description

Determines dual address preference.

### Notes

This property determines which address to give preference to if both a street address and a PO Box address are found and matched.

This property only comes into play if the two addresses are input as separate address lines. If they are input on the same address line, then USPS rules dictate that the PO Box address has priority, regardless of position.

The following table lists the available property values:

Value	Description
0	By position, giving preference to the bottom address when there are two address lines (default value)
1	Prefer PO Box address
2	Prefer street address

### CASS\_KEEP\_ALIAS\_ADDRESS

#### Data type

bool

#### Description

Determines how a street name input as an alias should be returned.

#### Notes

If property value is true, the input street name alias is returned. Otherwise, the "official" street name from the USPS database is returned.

The default value for this property is true.

### CASS\_KEEP\_EXTRA\_PRIMARY\_DATA

#### Data type

bool

#### Description

Determines if extra information in the primary address line is kept.

#### Notes

The default value for this property is false.

## CASS\_LIST\_NAME

### Data type

string

### Description

Specifies the name of the list to be printed on PS Form .

### Notes

The default value for this property is an empty string.

## CASS\_LIST\_PROCESSOR

### Data type

string

### Description

Specifies the name of the list processor to be printed on PS Form .

### Notes

The default value for this property is an empty string.

## CASS\_MAILERS\_ADDRESS

### Data type

string

### Description

Specifies the address of the mailer to be printed on PS Form .

### Notes

The default value for this property is an empty string.

## CASS\_MAILERS\_CITY

### Data type

string

### Description

Specifies the city of the mailer to be printed on PS Form .

**Notes**

The default value for this property is an empty string.

**CASS\_MAILERS\_NAME****Data type**

string

**Description**

Specifies the name of the mailer to be printed on PS Form .

**Notes**

The default value for this property is an empty string.

**CASS\_MAILERS\_STATE****Data type**

string

**Description**

Specifies the state of the mailer to be printed on PS Form .

**Notes**

The default value for this property is an empty string.

**CASS\_MAILERS\_ZIP****Data type**

string

**Description**

Specifies the ZIP Code of the mailer to be printed on PS Form .

**Notes**

The default value for this property is an empty string.

**CASS\_RETURN\_INPUT\_ADDRESS\_ON\_UNCONFIRMED\_DPV****Data type**

boolean

**Description**

Determines whether to roll back corrected addresses that are not DPV confirmable.

**Notes**

Read-only. The default value for this property is False.

When set to TRUE, addresses that cannot be verified for purposes of DPV are returned unchanged with casing applied, that is, address correction rolls back to the original address. The error code indicates why the address is not verified.

**CASS\_UPDATE\_UNCORRECTED\_CITY\_ST\_ZIP****Data type**

bool

**Description**

Determines if the city, state and ZIP Code data should be updated for an uncorrected address.

**Notes**

If property value is true, the city, state and ZIP Code data will be returned, when possible, for addresses that are not corrected when Update is called.

The default value for this property is false.

**CASS\_USE\_COUNTRY****Data type**

bool

**Description**

Use this property to incorporate the use of the Country field in processing foreign addresses.

Relies on data in the field FLD\_COUNTRY.

**Notes**

The default value for this property is False.

**CASS\_USE\_SUITELINK****Data type**

bool

## Description

This property is now obsolete. Suite<sup>Link</sup> is now a requirement of CASS processing.

## FORMAT\_CASING

## Data type

int

## Description

Determines the casing format applied to the address elements.

## Notes

The following table lists the available property values:

Value	Description
0	Upper case
1	Lower case
2	Mixed case (default value)

## FORMAT\_CITY

## Data type

int

## Description

Determines the format of the city name.

## Notes

The following table lists the available property values:

Value	Description
0	Return the full city name (default value)
1	Return abbreviated city name (if one exists)
2	Return as input THIS VALUE IS NO LONGER SUPPORTED

## FORMAT\_FIRM\_OUTPUT

### Data type

int

### Description

Determines the location of the firm name.

### Notes

This property determines how the business name is handled if it is not input as BUSINESS and is recognized as a firm name.

The following table lists the available property values:

Value	Description
0	Return as input (default value)
1	Return as BUSINESS (only if input BUSINESS was blank)
2	Always return as BUSINESS (will overwrite input BUSINESS)

## FORMAT\_HIGHWAY\_CONTRACT

### Data type

int

### Description

Determines the format of a highway contract address.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_PMB\_OUTPUT

### Data type

int

### Description

Determines the location of the private mailbox number.

### Notes

The following table lists the available property values:

Value	Description
0	Return on same line as unit information (default value)
1	Return as input

## FORMAT\_PO\_BOX

### Data type

int

### Description

Determines the format of a PO Box address.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_POST\_DIRECTIONAL

### Data type

int

### Description

Determines the format of the postdirectional element of an address.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_PRE\_DIRECTIONAL

### Data type

int

### Description

Determines the format of the predirectional element of an address.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_PRIMARY\_ADDRESS\_OUTPUT

### Data type

int

### Description

Determines the location of the primary address.

### Notes

The following table lists the available property values:

Value	Description
0	Return as ADDRESS_LINE_1 (if ADDRESS_LINE_2 is blank; otherwise, the primary address line will be returned as ADDRESS_LINE_2 with the secondary address data in ADDRESS_LINE_1); this is the default value
1	Return as ADDRESS_LINE_2

## FORMAT\_RURAL\_ROUTE

### Data type

int

### Description

Determines the format of a rural route address.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_SUFFIX

### Data type

int

### Description

Determines the format of the suffix element of an address.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_UNIT\_DESIGNATOR

### Data type

int

### Description

Determines the format of the unit designator.

### Notes

The following table lists the available property values:

Value	Description
0	Return standardized USPS abbreviation (USPS preferred format; this is also the default value)
1	Return standardized USPS abbreviation with punctuation
2	Return full word(s)

## FORMAT\_UNIT\_OUTPUT

### Data type

int

### Description

Determines the location of the unit information.

### Notes

The following table lists the available property values:

Value	Description
0	Return at end of primary address line (default value)
1	Return on secondary address line (if secondary address is blank)
2	Always return on secondary address line (existing secondary address data will be overwritten)

## FORMAT\_UPDATE\_CASE\_BUSINESS

### Data type

bool

### Description

Determines if the selected casing option is applied to the business name.

### Notes

Set to true to apply the selected casing option to the business name.

The default value for this property is true.

## FORMAT\_UPDATE\_CASE\_NAMES

### Data type

bool

### Description

Determines if the selected casing option is applied to the name fields.

## Notes

Set to true to apply the selected casing option to the first and last names.

The default value for this property is true.

## PRIMARY\_ADDRESS\_LENGTH\_LIMIT

### Data type

integer

### Description

**IMPORTANT** The PRIMARY\_ADDRESS\_LENGTH\_LIMIT property is designed to facilitate USPS CASS-certification testing and is not meant for general use. We strongly recommend that you use other abbreviation properties to configure primary address line abbreviation behavior.

Sets the maximum character length of the primary address line (address line 1). Abbreviation of different parts of the address line is determined by the settings of other abbreviation properties.

## Notes

It is not necessary to adjust the default setting unless you want to enforce that the abbreviation behavior use USPS street alias names where they exist. In that case, set this property to 31.

### Valid values

Default: 1000 (no limit)

31 (abbreviate to street name alias)

## SETTINGS\_CASS\_WIZARD\_CAPTION

### Data type

string

### Description

Specifies the caption that appears in the title bar of the CASS Wizard dialog.

## Notes

The default value of this property is "CASS Wizard."

## SETTINGS\_DATAFILE\_LOCATION

### Data type

string

### Description

Specifies the location of the `Address.cas` file.

### Notes

Setting this property will update the `Address.cas` path in the `mrtk.ini` file.

The default value for this property is an empty string.

## SETTINGS\_FIELD\_LIST\_IN

### Data type

USAddressFieldList

### Description

A USAddressFieldList object that defines which fields are supplied as input to the address-matching engine.

### Notes

This property defines the input fields that are contained in each USAddressRecord to be processed.

The input field list should contain, at a minimum, ADDRESS\_LINE\_1 plus either CITY/STATE or ZIP\_CODE or LAST\_LINE.

## SETTINGS\_FIELD\_LIST\_OUT

### Data type

USAddressFieldList

### Description

A USAddressFieldList object that defines which fields are returned as output from the address-matching engine.

### Notes

This property defines the output fields that are contained in each USAddressRecord that has been processed.

## SETTINGS\_HIDE\_PROGRESS\_AFTER\_BATCH

### Data type

bool

**Description**

Determines if the progress dialog remains visible after processing is complete.

**Notes**

Set to true to hide the progress dialog after processing is complete.

The CASS summary report can be previewed or printed from the progress dialog.

The default value for this property is false.

**SETTINGS\_INI\_FILE\_NAME****Data type**

string

**Description**

Specifies the full name and path of the ini file to use.

**Notes**

The default value for this property is "MRTK.INI."

Unless specified otherwise, the path is the Windows folder.

**SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT****Data type**

int

**Description**

Specifies the number of records contained in an address block.

**Notes**

This property specifies the number of USAddressRecord objects contained in a USAddressRecordBlock object.

The size of the record block determines the number of records processed with each call to Update.

**SETTINGS\_MAILROOM\_SERVER\_LIST****Data type**

string

### Description

Specifies the location of the BCC Architect Server.

### Notes

Setting this property creates a TCP/IP connection to the BCC Architect Server, which can reside on the local network or virtually anywhere.

This property should be set before calling PrepareTask.

We recommend that you use the BCC Architect Server when processing addresses from a Web site.

The format is: Server Name (or IP Address):Port.

Currently, going outside of the proxy server might not be supported.

The default value for this property is an empty string.

## SETTINGS\_MRTK\_VERSION

### Data type

string

### Description

Returns the current BCC Architect server version.

### Notes

The default value is an empty string.

## SETTINGS\_PRINT\_ON\_SERVER

### Data type

bool

### Description

Determines where reports are printed when using the BCC Architect Server.

### Notes

Set to true to print reports on the machine running BCC Architect Server; false to print on the client machine.

The default value for this property is false.

## SETTINGS\_RECORD\_COUNT

### Data type

int

### Description

Specifies the total number of records to be processed.

### Notes

The default value for this property is 0.

## SETTINGS\_REVIEW\_ERRORS

### Data type

bool

### Description

Determines whether to review uncoded records after all records have been processed.

### Notes

Set to true if ReviewErrors is to be called after all records have been processed with the Update function. ReviewErrors will display the Review Errors window, allowing the user to review and manually correct uncoded records.

The default value for this property is true.

## SETTINGS\_REVIEW\_ERRORS\_RECORD\_COUNT\_PER\_RECEIVE

### Data type

int

### Description

Specifies the number of records contained in the address block that is returned by each call to the RetrieveReviewed function.

### Notes

The default value for this property is 1.

## SETTINGS\_REVIEW\_ERRORS\_SHOW\_RECEIVE\_PROGRESS

### Data type

bool

### Description

Determines if the progress dialog is displayed while reviewed records are retrieved.

### Notes

Set to true to display the progress dialog while retrieving reviewed records with the RetrieveReviewed function.

The default value for this property is true.

## SETTINGS\_SHOW\_PROGRESS

### Data type

bool

### Description

Determines if the progress dialog is displayed during processing.

### Notes

Set to true to display the progress dialog during processing.

PS Form can be previewed or printed from the progress dialog after processing is complete.

The default value for this property is false.

## SETTINGS\_SILENT\_MODE

### Data type

bool

### Description

Determines CASSAssembly's mode of operation.

### Notes

Running CASSAssembly in silent mode disables all dialogs, including error messages.

The default value for this property is false.

## .NET CASSAssembly Fields Summary Table

The CASSAssembly fields are members of the USFields.CASS enumeration and are listed below. These enum names are used as arguments of the various functions of the USAddressFieldList and USAddressFields objects.

If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USFields.CASS enums in the Object Browser and IntelliSense.

Field ID	Description
ADDRESS_BLOCK	A block of addresses. Sets and retrieves the address as a complete block.
ADDRESS_LINE_1	The primary address line; required for CASS processing unless ADDRESS_LINE_2 is specified.
ADDRESS_LINE_2	The secondary address line; required for CASS processing unless ADDRESS_LINE_1 is specified.

Field ID	Description									
ADDRESS_SUGGESTION_LIST	<p>A list of suggested addresses that is returned based on an input address that is invalid and can't be corrected. Data is output to the field only if it is included in the output field list for batch address correction.</p> <p>The field will display a maximum of 10 entries, all of which have had their deliverability confirmed (DPV).</p> <p>By default, the following delimiters are used in the output text:</p> <table border="1"> <thead> <tr> <th>Delimiter Type</th> <th>ASCII</th> <th>Hex</th> </tr> </thead> <tbody> <tr> <td>Field</td> <td>25</td> <td>19</td> </tr> <tr> <td>Record</td> <td>26</td> <td>1A</td> </tr> </tbody> </table> <p>The type of input addresses that will generally return suggestions are those that have certain address elements that are missing or invalid, such as street suffixes and street directionals. Suggestions return the following fields:</p> <ul style="list-style-type: none"> <li>• AddressLine 1</li> <li>• City</li> <li>• State</li> <li>• ZIP Code</li> </ul>	Delimiter Type	ASCII	Hex	Field	25	19	Record	26	1A
Delimiter Type	ASCII	Hex								
Field	25	19								
Record	26	1A								
BUSINESS	The name of the business.									
CARRIER_ROUTE	The postal carrier route designation for the input record.									
CASSDATE	Contains information about when address was last processed.									
CITY	City; required for CASS processing unless LAST_LINE specified.									
CITY_ABBREVIATED	Abbreviated city of the input record after address correction and before NCOA processing.									
CONGRESSIONAL_DISTRICT	The congressional district number for this address.									

Field ID	Description
COUNTRY	Name of the Country.
COUNTY_CODE	5-digit county code for the input record.
COUNTY_NAME	Name of the County.
DP_BARCODE	Delivery Point barcode string for the input record after address correction and before NCOA processing.
DPC	DPC for the input record after address correction and before NCOA processing.
DPV_CODED	Indicates if the address was confirmed with DPV.
DPV_DOOR_NOT_ACCESSIBLE	Indicates addresses where carriers cannot knock on the door to deliver mail that will not fit into a mailbox or where carriers cannot physically access a residence/building.

Field ID	Description
DPV_FOOTNOTE	<p data-bbox="435 306 711 338">DPV footnote code:</p> <ul data-bbox="500 380 1484 1562" style="list-style-type: none"><li data-bbox="500 380 1442 411">• LK - Processing locked out due to a seed record being processed.</li><li data-bbox="500 457 951 489">• AA - Matched to the ZIP+4 file</li><li data-bbox="500 535 1024 567">• A1 - No match against the ZIP+4 file</li><li data-bbox="500 613 1263 644">• BB - Matched to DPV file (all components confirmed)</li><li data-bbox="500 690 1479 768">• CC - Matched only after removing secondary information; they were presented but invalid.</li><li data-bbox="500 814 1484 846">• N1 - Input primary matched, but high-rise missing secondary number.</li><li data-bbox="500 892 935 924">• M1 - Primary number missing.</li><li data-bbox="500 970 930 1001">• M3 - Primary number invalid.</li><li data-bbox="500 1047 1146 1079">• P1 - Input missing PO, RR or HC box number.</li><li data-bbox="500 1125 1386 1157">• P3 - Failed DPV because of invalid PO, RR or HC box number.</li><li data-bbox="500 1203 1114 1234">• RR - Matched CMRA (found in CMRA file).</li><li data-bbox="500 1281 1401 1312">• R1 - Matched CMRA, but secondary number (i.e., PMB) missing.</li><li data-bbox="500 1358 963 1390">• U1 - Matched unique ZIP Code.</li><li data-bbox="500 1436 954 1467">• G1 - Matched general delivery.</li><li data-bbox="500 1514 940 1545">• F1 - Matched military address.</li></ul>

Field ID	Description
DPV_INDICATOR	<p>Returns a single character that indicates the level of address validity with DPV:</p> <ul style="list-style-type: none"> <li>• Y - Both the primary and secondary (if present) validated against the DPV database.</li> <li>• S - The primary address is valid according to DPV, but the secondary is invalid.</li> <li>• D - The primary address is valid according to DPV, but the address is missing secondary information.</li> <li>• N - The primary address is not valid according to DPV.</li> <li>• "" - The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.</li> <li>• X - The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.</li> <li>• E - The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.</li> </ul>
DPV_IS_CMRA	Indicates if address is a commercial mail-receiving agent.
DPV_IS_PBSA	Indicates if address is a Post Office Box street address (PBSA).
DPV_IS_NOSTAT	Indicates if the address is not receiving delivery and is not counted as a possible delivery. The address is not receiving delivery because: 1) delivery has not been established, 2) The customer receives mail as part of a drop, 3) the carrier destroys or returns all of the mail.
DPV_IS_THROWBACK	Indicates if the address associated with the delivery point is a street address; however, the delivery is made to the customer's PO Box address.
DPV_IS_VACANT	Indicates if address is unoccupied.

Field ID	Description
DPV_NO_SECURE_LOCATION	Indicates the location of the address is not secure. The USPS can access the door, but cannot leave a package due to security concerns.
ERROR_CODE	Error code returned from address correction process.
ERROR_STRING	Text that explains the results from CASS processing.
EWS_CODED	Indicates that this record was flagged by the early warning system as a new address not yet included in the USPS databases.
EXTRA_INFO	Contains extra address information that the service was unable to parse into any other fields.
FIRST_NAME	First name.
GEOCODE_CENSUS_BLOCK	Census block – requires Geocode add-on.
GEOCODE_CENSUS_TRACT	Census tract – requires Geocode add-on.
GEOCODE_FOOTNOTE	Code indicating the granularity of ZIP Code match (ZIP Code length) for a geocode lookup – requires Geocode add-on: <ul style="list-style-type: none"> <li>• 00 - Unable to perform a Geocode lookup.</li> <li>• 03 - Geocode data based on a 3-digit ZIP Code.</li> <li>• 05 - Geocode data based on a 5-digit ZIP Code.</li> <li>• 07 - Geocode data based on a 7-digit ZIP Code.</li> <li>• 09 - Geocode data based on a 9-digit ZIP Code.</li> </ul>
GEOCODE_LATITUDE	Latitude coordinate – requires Geocode add-on.

Field ID	Description
GEOCODE_ LONGITUDE	Longitude coordinate – requires Geocode add-on.
GEOCODE_MSA_ CODE	Metropolitan statistical area code – requires Geocode add-on.
IS_RESIDENCE	Indicates if address is residential; requires RDI service subscription with USPS.
LACS_CODED	Indicates whether this address was flagged by LACS <sup>Link</sup> processing.
LACS_FOOTNOTE	Code indicating specific information about a LACS <sup>Link</sup> lookup: <ul style="list-style-type: none"> <li>• "" - Not processed / Seed record.</li> <li>• 00 - No match.</li> <li>• 09 - Matched to default high-rise address; address not updated.</li> <li>• 14 - Match failure to build new address.</li> <li>• 92 - Match secondary dropped from input.</li> <li>• A - Match success.</li> </ul>
LACS_INDICATOR	A single character indicating the result of a LACS <sup>Link</sup> lookup: <ul style="list-style-type: none"> <li>• "" - Not processed.</li> <li>• N - Match / Matched, but there was a failure to build new address.</li> <li>• Y - Match success.</li> <li>• S - Match with secondary dropped from input.</li> <li>• F - Seed record.</li> </ul>
LAST_LINE	Field containing city, state and ZIP Code; required for CASS processing unless CITY, STATE and ZIPCODE already specified.

Field ID	Description
LAST_NAME	Last name.
LOT_NUMBER	Line-of-Travel number.
MATCHED_TO_DEFAULT	TRUE if record should contain additional secondary address information (but is not required to).
PMB_NUMBER	Private mailbox number.
PO_BOX_DELIVERY_ONLY_ZIP	Indicates whether delivery is to an address in a PO Box only zone.
POST_DIRECTIONAL	Post-directional address element.
PRE_DIRECTIONAL	Pre-directional address element.
PRIMARY_NUMBER	Primary number address element.
RECORD_ID	User field that can contain the input record's index or ID.
RECORD_TYPE	The following record types are currently valid: <ul style="list-style-type: none"> <li>• S – Street record</li> <li>• P – Post office box</li> <li>• R – Rural Route or Highway Contract</li> <li>• H – High-rise, Building or Apartment</li> <li>• F – Firm Record</li> <li>• G – General Delivery</li> </ul>
SKIPPED_CERTIFY	TRUE if record was skipped during CASS processing.

Field ID	Description
STATE	State; required for CASS processing unless LAST_LINE specified.
STREET_NAME	Street name address element.
SUFFIX	Suffix address element.
SUITE_LINK_ FOOTNOTE	<p>Suite<sup>Link</sup> footnote code:</p> <ul style="list-style-type: none"> <li>• "" – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup within the Suite<sup>Link</sup> file. Only default high-rise addresses qualify for a Suite<sup>Link</sup> lookup.</li> <li>• A – The address was processed and secondary information was added to the resulting address.</li> <li>• 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.</li> </ul>
UNIT_ DESIGNATOR	Unit designator address element.
UNIT_NUMBER	Unit number address element.
URBANIZATION	Urbanization number. Puerto Rico addresses only.
USER_DEFINED_1	Custom data to attach to a record.
USER_DEFINED_2	Custom data to attach to a record.
USER_DEFINED_3	Custom data to attach to a record.
USER_DEFINED_4	Custom data to attach to a record.
USER_DEFINED_5	Custom data to attach to a record.
USER_DEFINED_6	Custom data to attach to a record.
USER_DEFINED_7	Custom data to attach to a record.

Field ID	Description
USER_DEFINED_8	Custom data to attach to a record.
USER_DEFINED_9	Custom data to attach to a record.
USER_DEFINED_10	Custom data to attach to a record.
USER_DEFINED_11	Custom data to attach to a record.
USER_DEFINED_12	Custom data to attach to a record.
USER_DEFINED_13	Custom data to attach to a record.
USER_DEFINED_14	Custom data to attach to a record.
USER_DEFINED_15	Custom data to attach to a record.
ZIP_CODE	The ZIP Code of this address.
ZIP4_FOOTNOTE	ZIP4 footnote code. See DPV_FOOTNOTE for code descriptions.

## .NET CASSAssembly Reports Summary Table

The CASSAssembly reports are members of the USReports.CASS enumeration and are defined below. These enum names are used as arguments of the various functions to preview, print or save a report. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USReports.CASS enums in the Object Browser and IntelliSense.

Name	Description
CASS_FORM	CASS Summary Report (PS Form ); this report is required for automation sorts.

## The .NET MOVEAssembly Class for Updating Moved Addresses

The BCC Architect MOVEAssembly object processes a list of addresses and updates it with the latest National Change Of Address (NCOA<sup>Link</sup>) data.

The process occurs over three phases. During the first phase, the addresses are sent in blocks to the Move Update Server, which is usually located at BCC Software, but can be located in-house as well. The addresses are also corrected during phase one. During the second phase, the list is compared against the NCOA<sup>Link</sup> database, and records that are matched are updated to the new address. The list is retrieved, again in blocks, during the third and final phase. After the job is finished, you can print the CASS Summary Report (PS Form ) and NCOA<sup>Link</sup> Summary Report.

MOVEAssembly provides a flexible interface through which you can control the amount of information returned for each address as well as the number of records that are sent or retrieved with each block.

### Using .NET MOVEAssembly

The following general procedure should be used to implement the MOVEAssembly class:

1. Add a reference to the .NET component Satori.MRTK.USAssembly.
2. While not required, you can reduce the amount of typing needed by adding using statements (C#) or Imports statements (Visual Basic) for the Satori.MailRoomToolKit.US and Satori.MailRoomToolKit.US Interfaces namespaces.
3. Create a MOVEAssembly object.
4. Call SetProperty using SETTINGS\_MAILROOM\_SERVER\_LIST to set the location of the Move Update server.
5. Call PrepareTask.
6. Create a USAddressFieldList object that defines the input fields for each record. Fields are added to the USAddressFieldList object using the USFields.Field enumeration. The minimum amount of address data required is ADDRESS\_LINE\_1 plus either ZIP\_CODE, CITY and STATE, or LAST\_LINE. In addition, the FIRST\_NAME and LAST\_NAME fields are required for individual and family name matching and the BUSINESS field is required for business matching. For best results, it is recommended that all of the basic name and address fields be used: FIRST\_NAME, LAST\_NAME, BUSINESS, ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE, and ZIP\_CODE (or LAST\_LINE). Alternatively, the entire address (not including names) can be passed in at once with the ADDRESS\_BLOCK field.

7. Create a `USAddressFieldList` object that defines the updated output fields for each processed record. Typically, this would include the basic address fields plus any of the data fields that result from the move update process that are of use to you (e.g., `MOVE_EFFECTIVE` or `MOVE_TYPE`).
8. Call `SetProperty` to set the desired properties. The individual properties are specified using the `USProperties.Move` enumeration. You must set `SETTINGS_FIELD_LIST_IN` and `SETTINGS_FIELD_LIST_OUT` using the `USAddressFieldList` objects that were created in steps 6 and 7, respectively. You should also specify the number of records to be processed at a time with `SETTINGS_INPUT_BLOCK_RECORD_COUNT`.
9. Call `ValidateProperties`.
10. Create a `USAddressRecordBlock` object.
11. Loop through the records in your database, and for each record to be processed:
12. Create a `USAddressRecord` object. A `USAddressFieldList` object is required when creating a `USAddressRecord` object. Use the `USAddressFieldList` object created in step 6.
13. Set the input field values using the `Fields` property of the `USAddressRecord` object.
14. Add the `USAddressRecord` object to the `USAddressRecordBlock` object.
15. Repeat steps a through c until the `USAddressRecordBlock` object contains `SETTINGS_INPUT_BLOCK_RECORD_COUNT` `USAddressRecord` objects.
16. Call `Send`, passing in the `USAddressRecordBlock` object.
17. Repeat steps and until all records have been sent. Call the `Clear` function of the `USAddressRecordBlock` object to reset it before adding a new block of records to it.
18. Call `DoProcess`.
19. Call `Retrieve`. A new, updated `USAddressRecordBlock` object will be returned.
20. Iterate through the `USAddressRecordBlock` object to get each updated `USAddressRecord` object. In turn, iterate through each `USAddressRecord` object to get the updated fields for each record. The fields for each output record will match those specified by the `SETTINGS_FIELD_LIST_OUT` property set earlier.

21. Repeat steps and until all records have been retrieved.
22. Call `PreviewMoveReports` or `PrintMoveReports` to preview or print the CASS Summary Report (PS Form ) and NCOA<sup>Link</sup> Processing Summary Report. You can also save the reports as a PDF file using the `SaveMoveReportsAsPDF` function.
23. Call `EndTask`.

## .NET MOVEAssembly Functions

The MOVEAssembly functions are defined below. If you have added the `Satori.MRTK.USAssembly` reference to your .NET project, then you can view all of the available MOVEAssembly members, and their definitions, in the Object Browser and IntelliSense.

### DoProcess

#### Syntax

```
void DoProcess();
```

#### Description

Performs NCOA<sup>Link</sup> processing on all records sent to the Move Update Server.

#### Parameters

None.

#### Return values

None.

#### Notes

You should call this function after your final call to `Send`.

A call to this function will initiate the move update process, and all records sent to the Move Update Server will be compared against the NCOA<sup>Link</sup> database.

#### See also

[Send](#)

[DoProcessEx](#)

## DoProcessEx

### Syntax

```
bool DoProcessEx();
```

### Description

Performs NCOA<sup>Link</sup> processing on the records sent to the Move Update Server, in blocks of 0 records at a time.

### Parameters

None.

### Return values

true

Indicates some of the records sent have not yet been processed.

false

Indicates all records have been processed.

### Notes

You should call this function after your final call to `Send`.

A call to this function will initiate the move update process, and the records sent to the Move Update Server will be compared against the NCOA<sup>Link</sup> database.

Unlike `DoProcess`, which processes the entire list sent before returning execution to the calling program, this function will process the list in increments of 0 records, so you must use a loop to continuously call it until the returned value is false.

### See also

[Send](#)

[DoProcess](#)

## EndTask

### Syntax

```
void EndTask();
```

### Description

Cleans up and releases a MOVEAssembly object.

### Parameters

None.

### Return values

None.

### Notes

EndTask cleans up and releases used resources and should be called when done with the MOVEAssembly object.

## GetProperty

### Syntax

```
string GetProperty(USProperties.Move movePropertyId);
```

### Description

Retrieves the current value of a MOVEAssembly property.

### Parameters

*movePropertyId*

Specifies the enum name of the property to get.

### Return values

The value of the property specified by movePropertyId.

### Notes

All property values are returned as a string, regardless of the data type passed into SetProperty.

### See also

See the [MOVEAssembly Properties](#) section for a complete list of properties.

## PrepareTask

### Syntax

```
void PrepareTask();
```

## Description

Initializes and prepares the MOVEAssembly object.

## Parameters

None.

## Return values

None.

## Notes

PrepareTask should be called only once, after the MOVEAssembly object is created.

This function must be called before calling any of the other functions or setting any of the properties of MOVEAssembly. Failing to do so will cause subsequent function calls to fail. The one exception to this rule is setting the SETTINGS\_MAILROOM\_SERVER\_LIST property, which must be defined prior to calling PrepareTask.

## PreviewMoveReports

### Syntax

```
void PreviewMoveReports();
```

### Description

Previews all Move Update reports.

### Parameters

None.

### Return values

None.

### See also

[PreviewReport](#)

## PreviewReport

### Syntax

```
void PreviewReport(USReports.Move mrtkReportId);
```

## Description

Previews a Move Update report.

## Parameters

*mrtkReportId*

The ID of the report to preview.

## Return values

None.

## Notes

Unlike `PreviewMoveReports`, which previews all of the Move Update reports, this function only displays the specified report.

## See also

[PreviewMoveReports](#)

See the [MOVEAssembly Reports](#) section for a complete list of reports.

## PrintMoveReports

### Syntax

```
void PrintMoveReports(string printerName, bool showPrintSetupDialog);
```

### Description

Prints all Move Update reports.

### Parameters

*printerName*

The name of the printer you wish to print to.

*showPrintSetupDialog*

Set to true to show the printer setup dialog or false to hide this dialog.

### Return values

None.

## Notes

If `printerName` is set to an empty string then the default printer will be used to print the form.

If `showPrintSetupDialog` is set to true then the value for `printerName` will be ignored.

If the printer is a network printer, the `printerName` parameter must be set to the full name (e.g., \\Server\HP Laser Jet 5). To find the full printer name, you could print a test page from the print driver.

## See also

### [PrintReport](#)

## PrintReport

### Syntax

```
void PrintReport(USReports.Move mrtkReportId, string printerName,  
bool showPrintSetupDialog);
```

### Description

Prints a Move Update report.

### Parameters

*mrtkReportId*

The ID of the report to print.

*printerName*

The name of the printer you wish to print to.

*showPrintSetupDialog*

Set to true to show the printer setup dialog or false to hide this dialog.

### Return values

None.

## Notes

If `printerName` is set to an empty string then the default printer will be used to print the form.

If `showPrintSetupDialog` is set to true then the value for `printerName` will be ignored.

If the printer is a network printer, the `printerName` parameter must be set to the full name (e.g., `\\Server\HP Laser Jet 5`). To find the full printer name, you could print a test page from the print driver.

Unlike `PrintMoveReports`, which prints all of the Move Update reports, this function only prints the specified report.

See also

### [PrintMoveReports](#)

See the [MOVEAssembly Reports](#) section for a complete list of reports.

## Retrieve

### Syntax

```
USAddressRecordBlock Retrieve();
```

### Description

Retrieves the processed records from the Move Update Server.

### Parameters

None.

### Return values

A `USAddressRecordBlock` object containing the updated addresses.

### Notes

The number of records in the returned record block is determined by the `SETTINGS_INPUT_BLOCK_RECORD_COUNT` property.

The fields in each `USAddressRecord` object in the address block are determined by the `SETTINGS_FIELD_LIST_OUT` property.

See also

See the [MOVEAssembly Properties](#) section for the definition of:

[SETTINGS\\_FIELD\\_LIST\\_OUT](#)

[SETTINGS\\_INPUT\\_BLOCK\\_RECORD\\_COUNT](#)

## SaveMoveReportsAsPDF

### Syntax

```
string SaveMoveReportsAsPDF(string fileName, int fileOption);
```

### Description

Saves all Move Update reports.

### Parameters

#### *filename*

The name, including path, of the file to save.

#### *fileOption*

Specifies what to do if the supplied file name already exists:

- 0 – Overwrite if file with file name specified in fileName already exists.
- 1 – Prompt if file with file name specified in fileName already exists.
- 2 – Create new file if file with name specified in fileName already exists; a number will be appended to the file name.
- 3 – Append if file with name specified in fileName already exists (not yet implemented).

### Return values

Returns a string containing the path and name of the saved file.

### See also

[SaveReportAsPDF](#)

## SaveReportAsPDF

### Syntax

```
string SaveReportAsPDF(USReports.Move mrTkReportId, string fileName,  
int fileOption);
```

### Description

Saves a report as a PDF file.

## Parameters

*mrtkReportId*

The ID of the report to save.

*filename*

The name, including path, of the file to save.

*fileOption*

Specifies what to do if the supplied file name already exists:

- 0 – Overwrite if file with file name specified in fileName already exists.
- 1 – Prompt if file with file name specified in fileName already exists.
- 2 – Create new file if file with name specified in fileName already exists; a number will be appended to the file name.
- 3 – Append if file with name specified in fileName already exists (not yet implemented).

## Return values

Returns a string containing the path and name of the saved file.

## Notes

Unlike `SaveMoveReportsAsPDF`, which saves all of the Move Update reports, this function only saves the specified report.

## See also

[SaveMoveReportsAsPDF](#)

See the [MOVEAssembly Reports](#) section for a complete list of reports.

## Send

### Syntax

```
void Send(USAddressRecordBlock addressBlock);
```

### Description

Sends an address block to the Move Update server.

## Parameters

*addressBlock*

A USAddressRecordBlock object containing the input records to be processed.

## Return values

None.

## Notes

You may want to experiment with the number of records in an input block as specified by the `SETTINGS_INPUT_BLOCK_RECORD_COUNT` property. In preliminary tests, we have found the optimal setting to be around 25 – 50.

The fields in each USAddressRecord object in the address block are determined by the `SETTINGS_FIELD_LIST_IN` property.

You should call `DoProcess` after you have finished sending all of your records to the Move Update Server.

## See also

[DoProcess](#)

See the [MOVEAssembly Properties](#) section for the definition of:

[SETTINGS\\_FIELD\\_LIST\\_IN](#)

[SETTINGS\\_INPUT\\_BLOCK\\_RECORD\\_COUNT](#)

## SetProperty

### Syntax

```
void SetProperty(USProperties.CASS cassPropertyId,  
object val)
```

```
void SetProperty(USProperties.Move movePropertyId,  
object val)
```

### Description

Sets the value of a MOVEAssembly property.

#### Parameters

*cassPropertyId*

Specifies the enum name of the CASS property to set.

*movePropertyId*

Specifies the enum name of the Move Update property to set.

*val*

Specifies the new property value.

#### Return values

None.

#### Notes

This function is overloaded to accept both CASS and Move Update properties.

#### See also

[GetProperty](#)

[ShowMoveWizard](#)

See the [CASSAssembly Properties](#) section for a complete list of CASS properties.

See the [MOVEAssembly Properties](#) section for a complete list of Move Update properties.

## ShowMoveWizard

#### Syntax

```
void ShowMoveWizard();
```

#### Description

Displays the Move Wizard.

#### Parameters

None.

#### Return values

None.

## Notes

Call this function if you want to display the Move Update Wizard. The Move Update Wizard provides a graphical interface that leads a user through the steps necessary to process a list with the NCOA<sup>Link</sup> service.

Properties set through the wizard do not have to be set through SetProperty.

## See also

[SetProperty](#)

## ValidateProperties

### Syntax

```
void ValidateProperties();
```

### Description

Verifies that the MOVEAssembly object is set up correctly and ready to run.

### Parameters

None.

### Return values

None.

## Notes

This function verifies that the basic requirements of a MOVEAssembly object have been met.

This function needs to be called after setting property values and before calling Send.

## See also

[SetProperty](#)

## .NET MOVEAssembly Properties

The MOVEAssembly properties are members of the USProperties.Move enumeration and are defined below. These enum names are used as arguments of the GetProperty and SetProperty functions. These two functions are overloaded to also accept properties from the USProperties.CASS enumeration. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then

you can view all of the available USProperties.Move and USProperties.CASS enums in the Object Browser and IntelliSense.

## KEEP\_ALTERNATE\_ADDRESS\_LINE

### Data type

bool

### Description

Specifies whether to retain original address information in addition to the updated address when a moved address is found.

### Notes

If True, retains original address information in addition to the updated address when a moved address is found.

Default value is False

## LOGIN\_ADMIN\_ID

### Data type

string

### Description

Specifies the NCOA<sup>Link</sup> User ID for a List Administrator account type.

### Notes

The default value for this property is an empty string.

## LOGIN\_ADMIN\_PASSWORD

### Data type

string

### Description

Specifies the NCOA<sup>Link</sup> password for a List Administrator account type.

### Notes

The default value for this property is an empty string.

## LOGIN\_BROKER\_ID

### Data type

string

### Description

Specifies the NCOA<sup>Link</sup> User ID for a Broker account type.

### Notes

The default value for this property is an empty string.

## LOGIN\_BROKER\_PASSWORD

### Data type

string

### Description

Specifies the NCOA<sup>Link</sup> password for a Broker account type.

### Notes

The default value for this property is an empty string.

## LOGIN\_CUSTOMER\_ID

### Data type

string

### Description

Specifies the NCOA<sup>Link</sup> User ID for a Client account type.

### Notes

The default value for this property is an empty string.

## LOGIN\_CUSTOMER\_PASSWORD

### Data type

string

### Description

Specifies the NCOA<sup>Link</sup> password for a Client account type.

**Notes**

The default value for this property is an empty string.

**MOVE\_BUYER\_NAME****Data type**

string

**Description**

Specifies the name of the person who bought the list processed with NCOA<sup>Link</sup>.

**Notes**

The default value for this property is an empty string.

**MOVE\_CLIENT\_ID\_LIST****Data type**

String

**Description**

Returns a list of client IDs and their PAF expiration dates as a single string.

**Notes**

The format is [CLIENT\_ID],YYYY-MM-DD. Each ID-date pair is separated from others by a carriage return.

Before you can retrieve this data, you must set LOGIN\_BROKER\_ID or LOGIN\_ADMIN\_ID first, then call ValidateProperties.

The default value is an empty string.

**MOVE\_CLIENT\_ID\_NAME\_LIST****Data type**

String

**Description**

Returns a list of client IDs, their names, their PAF expiration dates, and the status of their PAF form as a single string.

Notes

The format is [CLIENT\_ID], CUSTOMER\_NAME, MM-DD-YYYY, STATUS. Each ID-name-date-status entry is separated from others by a carriage return.

STATUS can have one of the following values:

- Expired – PAF is expired.
- Pending – PAF signature is pending approval.
- Blank – PAF is up-to-date.

Before you can retrieve this data, you must set LOGIN\_BROKER\_ID or LOGIN\_ADMIN\_ID first, then call Validate- Properties.

The default value is an empty string.

MOVE\_HIGH\_MATCH\_RATE\_REASON

Data type

string

Description

Specifies information about the list processed with NCOA<sup>Link</sup>.

Notes

The value of this property is stored in the NCOA<sup>Link</sup> job table along with other statistics about the job.

The following table lists the available property values:

Value	Description
A	The input file consists of packages that were matched beyond months in an ANK <sup>Link</sup> file
S	The input file is a Stage 1 file
R	The input file consists of addresses that generated returned mail
""	No reason given (default value)

## MOVE\_MAIL\_CLASS

### Data type

string

### Description

Specifies the class of mailing for the list processed with NCOA<sup>Link</sup>.

### Notes

The following table lists the available property values:

Value	Description
A	First Class only
B	Periodicals only
C	Standard Mail only
D	Package Services only
E	First Class and Periodicals
F	First Class and Standard Mail
G	First Class and Package Services
H	Periodicals and Standard Mail
I	Periodicals and Package Services
J	Standard Mail and Package Services
K	First Class, Periodicals, and Standard Mail
L	First Class, Periodicals, and Package Services
M	First Class, Standard Mail, and Package Services
N	Periodicals, Standard Mail, and Package Services
O	All (default value)

## MOVE\_MATCH\_FLAG

### Data type

string

### Description

Determines the type of moves the service will search for.

### Notes

The following table lists the available property values:

Value	Description
S	Standard. Business, Individual and Family. This is the default value.
C	Business and Individual.
B	Business only.
I	Individual only.
R	Individual and Family.

## MOVE\_MOVE\_MONTH\_RANGE

### Data type

int

### Description

Determines the maximum number of months ago a move may have occurred in order for the address change to be applied.

### Notes

Moves that are older than the value specified by this property are identified but the address will not be updated.

The minimum value for this property is 6.

The maximum value is 18 for a limited service provider and 48 for a full service provider. BCC Software is a full service provider.

The default value for this property is 48.

## MOVE\_MULTI\_NAME\_HANDLE

Data type

int

Description

Determines how addresses with multiple names are handled.

Notes

The following table lists the available property values:

Value	Description
1	Search only if a common last name is found
3	Search using all names found
0	Skip the address

## NCOA\_CUSTOMER\_ADDRESS

Data type

string

Description

The address for the client customer processing the list.

## NCOA\_CUSTOMER\_EMAIL

Data type

string

Description

The email address for the client customer processing the list.

## NCOA\_CUSTOMER\_FAX

Data type

string

## Description

The fax number for the client customer processing the list.

**NCOA\_CUSTOMER\_LASTLINE**

## Data type

string

## Description

The last line of the address for the client customer processing the list.

**NCOA\_CUSTOMER\_NAME**

## Data type

string

## Description

The name of the client customer processing the list.

**NCOA\_CUSTOMER\_PHONE**

## Data type

string

## Description

The phone number for the client customer processing the list.

**NCOA\_CUSTOMER\_SIC**

## Data type

string

## Description

The Standard Industrial Classification (now called NAICS) code for the client customer processing the list.

**NCOA\_CUSTOMER\_TITLE**

## Data type

string

**Description**

The job title for the client customer processing the list.

**SETTINGS\_FIELD\_LIST\_IN****Data type**

USAddressFieldList

**Description**

A USAddressFieldList object that defines which fields are supplied as input to the address-matching engine.

**Notes**

This property defines the input fields that are contained in each USAddressRecord to be processed.

The input field list should contain, at a minimum, ADDRESS\_LINE\_1 plus either CITY/STATE or ZIP\_CODE or LAST\_LINE. In addition, the FIRST\_NAME and LAST\_NAME fields are required for individual and family name matching and the BUSINESS field is required for business matching.

**SETTINGS\_FIELD\_LIST\_OUT****Data type**

USAddressFieldList

**Description**

A USAddressFieldList object that defines which fields are returned as output from the address-matching engine.

**Notes**

This property defines the output fields that are contained in each USAddressRecord that has been processed.

**SETTINGS\_HIDE\_PROGRESS\_AFTER\_PROCESS****Data type**

bool

**Description**

Determines if the Progress dialog box remains visible after processing is complete.

## Notes

Set to true to hide the Progress dialog box after processing is complete.

The Move Update reports can be previewed or printed from the progress dialog.

The default value for this property is false.

## SETTINGS\_INI\_FILE\_NAME

### Data type

string

### Description

Specifies the full name and path of the .ini file to use.

## Notes

The default value for this property is **MRTK.INI**.

Unless specified otherwise, the path is the Windows folder.

## SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT

### Data type

int

### Description

Specifies the number of records contained in an address block.

## Notes

This property specifies the number of USAddressRecord objects contained in a USAddressRecordBlock object.

The size of the record block determines the number of records sent to the Move Update server with each call to Send and the number of records returned by the Retrieve function.

## SETTINGS\_MAILROOM\_SERVER\_LIST

### Data type

string

### Description

Specifies the location of the BCC Architect Server.

## Notes

Setting this property creates a TCP/IP connection to the Move Update Server.

This property should be set before calling PrepareTask.

The format is: Server Name (or IP Address):Port.

The default value for this property is an empty string.

## SETTINGS\_PRINT\_ON\_SERVER

### Data type

bool

### Description

Determines where reports are printed.

## Notes

Set to true to print the Move Update reports on the machine running the Move Update Server; false to print on the client machine.

The default value for this property is false.

## SETTINGS\_MRKT\_VERSION

### Data type

string

### Description

Returns the current BCC Architect server version.

## Notes

The default value is an empty string.

## SETTINGS\_SHOW\_PROGRESS

### Data type

bool

### Description

Determines if the Progress dialog box is displayed during processing.

## Notes

Set to true to display the progress dialog during processing.

The Move Update reports can be previewed or printed from the Progress dialog box after processing is complete.

The default value for this property is false.

## .NET MOVEAssembly Reports

The MOVEAssembly reports are members of the USReports.MOVE enumeration and are defined below. These enum names are used as arguments of the various functions to preview, print or save a report. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USReports.MOVE enums in the Object Browser and IntelliSense.

Name	Description
CASS_FORM	CASS Summary Report (PS Form ); this report is required for automation sorts.
SUMMARY	NCOA <sup>Link</sup> Processing Summary Report.

## .NET MOVEAssembly Fields

The MOVEAssembly fields are members of the USFields.Move enumeration and are listed below. These enum names are used as arguments of the various functions of the USAddressFieldList and USAddressFields objects. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USFields.Move enums in the Object Browser and IntelliSense.

Field ID	Description
ADDRESS_BLOCK	A block of addresses. Sets and retrieves the address as a complete block.
ADDRESS_LINE_1	The primary address line; required for CASS processing unless ADDRESS_LINE_2 is specified.
ADDRESS_LINE_2	The secondary address line; required for CASS processing unless ADDRESS_LINE_1 is specified.
AFTER_CASS_ADDRESS_BLOCK	Address Block of the input record after address correction and before NCOA <sup>Link</sup> processing.

Field ID	Description
AFTER_CASS_ADDRESS_LINE_1	Address Line 1 of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_ADDRESS_LINE_2	Address Line 2 of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_CARRIER_ROUTE	Carrier route for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_CASSDATE	Contains information about when the input record, after address correction and before NCOA <sup>Link</sup> processing, was last processed.
AFTER_CASS_CITY	City of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_CITY_ABBREVIATED	Abbreviated city of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_CONGRESSIONAL_DISTRICT	The congressional district number for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_COUNTY_CODE	County code for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_COUNTY_NAME	County name for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_DP_BARCODE	Delivery Point barcode string for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_DPC	DPC for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_DPV_CODED	Indicates if the address after address correction and before NCOA <sup>Link</sup> processing was matched to a known delivery point.

Field ID	Description
AFTER_CASS_DPV_DOOR_NOT_ACCESSIBLE	After address correction and before NCOA <sup>Link</sup> processing, returns a DPV code that indicates that the door is not physically accessible by mail carrier.
AFTER_CASS_DPV_INDICATOR	Indicates the results of DPV processing after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_DPV_IS_CMRA	Indicates if the address, after address correction and before NCOA <sup>Link</sup> processing, is a Commercial Mail Receiving Agent.
AFTER_CASS_DPV_IS_THROWBACK	After address correction and before NCOA <sup>Link</sup> processing, returns a DPV code that indicates that delivery is made to a PO Box address.
AFTER_CASS_DPV_IS_VACANT	Indicates if the input record, after address correction and before NCOA <sup>Link</sup> processing, is unoccupied.
AFTER_CASS_DPV_NO_SECURE_LOCATION	After address correction and before NCOA <sup>Link</sup> processing, returns a DPV code that indicates that the location is not secure.
AFTER_CASS_ERROR_CODE	Error code for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_ERROR_STRING	Error string for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_EWS_CODED	Indicates if the address, after address correction and before NCOA <sup>Link</sup> processing, was coded for the Early Warning System (EWS) for new development or construction.
AFTER_CASS_IS_RESIDENCE	Indicates if the address, after address correction and before NCOA <sup>Link</sup> processing, is a residential address.
AFTER_CASS_LACS_CODED	Indicates if the address, after address correction and before NCOA <sup>Link</sup> processing, was coded for the Locatable Address Conversion System (LACS).

Field ID	Description
AFTER_CASS_ LACS_FOOTNOTE	Code indicating specific information about a LACS <sup>Link</sup> lookup for the input record after address correction and before NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"> <li>• "" - Not processed / Seed record.</li> <li>• 00 - No match.</li> <li>• 09 - Matched to default high-rise address; address not updated.</li> <li>• 14 - Match failure to build new address.</li> <li>• 92 - Match secondary dropped from input.</li> <li>• A - Match success.</li> </ul>
AFTER_CASS_ LACS_INDICATOR	A single character indicating the result of a LACS <sup>Link</sup> lookup for the input record after address correction and before NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"> <li>• "" - Not processed.</li> <li>• N - Match / Matched, but there was a failure to build new address.</li> <li>• Y - Match success.</li> <li>• S - Match with secondary dropped from input.</li> <li>• F - Seed record.</li> </ul>
AFTER_CASS_ LAST_LINE	The combined City, State, and ZIP code for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_ LOT_NUMBER	The Line of Travel number for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_ MATCHED_TO_ DEFAULT	TRUE if the input record, after address correction and before NCOA <sup>Link</sup> processing, should contain additional secondary address information (but is not required to).

Field ID	Description
AFTER_CASS_PMB_NUMBER	The Private Mail Box number of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_POST_DIRECTIONAL	Post-directional of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_PO_BOX_DELIVERY_ONLY_ZIP	Indicates whether delivery is to an address in a PO Box only zone, after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_PRE_DIRECTIONAL	Predirectional of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_PRIMARY_NUMBER	Primary number of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_RECORD_TYPE	Record type for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_STATE	State for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_STREET_NAME	Street name of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_SUFFIX	Suffix of the input record after address correction and before NCOA <sup>Link</sup> processing.

Field ID	Description
AFTER_CASS_SUITELINK_FOOTNOTE	<p>The Suite<sup>Link</sup> footnote code for the input record after address correction and before NCOA<sup>Link</sup> processing:</p> <ul style="list-style-type: none"> <li>• "" – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup within the Suite<sup>Link</sup> file. Only default high-rise addresses qualify for a Suite<sup>Link</sup> lookup.</li> <li>• A – The address was processed and secondary information was added to the resulting address.</li> <li>• 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.</li> </ul>
AFTER_CASS_UNIT_DESIGNATOR	Unit designator of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_UNIT_NUMBER	Unit number of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_URBANIZATION	Urbanization of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_USPS_4_STATE_BARCODE	The Intelligent Mail barcode for the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_CASS_ZIP_CODE	ZIP Code of the input record after address correction and before NCOA <sup>Link</sup> processing.
AFTER_NCOA_ADDRESS_BLOCK	Address Block of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_ADDRESS_LINE_1	Address Line 1 of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_ADDRESS_LINE_2	Address Line 2 of the input record after NCOA <sup>Link</sup> processing.

Field ID	Description
AFTER_NCOA_CARRIER_ROUTE	Carrier route for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_CASSDATE	Contains information about when the input record, after NCOA <sup>Link</sup> processing, was last processed.
AFTER_NCOA_CITY	City of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_CITY_ABBREVIATED	Abbreviated city of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_CONGRESSIONAL_DISTRICT	The congressional district number for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_COUNTY_CODE	County code for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_COUNTY_NAME	County name for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_DP_BARCODE	Delivery Point barcode string for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_DPC	DPC for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_DPV_CODED	Indicates if the address after NCOA <sup>Link</sup> processing was matched to a known delivery point.
AFTER_NCOA_DPV_INDICATOR	Indicates the results of DPV processing after NCOA <sup>Link</sup> processing.
AFTER_NCOA_DPV_IS_CMRA	Indicates if the address, after NCOA <sup>Link</sup> processing, is a Commercial Mail Receiving Agent.
AFTER_NCOA_DPV_IS_VACANT	Indicates if the input record, after NCOA <sup>Link</sup> processing, is unoccupied.

Field ID	Description
AFTER_NCOA_ERROR_CODE	Error code for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_ERROR_STRING	Error string for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_EWS_CODED	Indicates if the address, after NCOA <sup>Link</sup> processing, was coded for the Early Warning System (EWS) for new development or construction.
AFTER_NCOA_IS_RESIDENCE	Indicates if the address, after NCOA <sup>Link</sup> processing, is a residential address.
AFTER_NCOA_LACS_CODED	Indicates if the address, after NCOA <sup>Link</sup> processing, was coded for the Locatable Address Conversion System (LACS).
AFTER_NCOA_LACS_FOOTNOTE	Code indicating specific information about a LACS <sup>Link</sup> lookup for the input record after NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"> <li>• "" - Not processed / Seed record.</li> <li>• 00 - No match.</li> <li>• 09 - Matched to default high-rise address; address not updated.</li> <li>• 14 - Match failure to build new address.</li> <li>• 92 - Match secondary dropped from input.</li> <li>• A - Match success.</li> </ul>

Field ID	Description
AFTER_NCOA_LACS_INDICATOR	<p>A single character indicating the result of a LACS<sup>Link</sup> lookup for the input record after NCOA<sup>Link</sup> processing:</p> <ul style="list-style-type: none"> <li>• "" - Not processed.</li> <li>• N - Match / Matched, but there was a failure to build new address.</li> <li>• Y - Match success.</li> <li>• S - Match with secondary dropped from input.</li> <li>• F - Seed record.</li> </ul>
AFTER_NCOA_LAST_LINE	The combined City, State, and ZIP code for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_LOT_NUMBER	The Line of Travel number for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_MATCH_FIRST_NAME	
AFTER_NCOA_MATCH_LAST_NAME	
AFTER_NCOA_MATCH_SUFFIX_NAME	
AFTER_NCOA_MATCHED_TO_DEFAULT	TRUE if the input record, after NCOA <sup>Link</sup> processing, should contain additional secondary address information (but is not required to).
AFTER_NCOA_MIDDLE_INITIAL	

Field ID	Description
AFTER_NCOA_PMB_NUMBER	The Private Mail Box number of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_PO_BOX_DELIVERY_ONLY_ZIP	Indicates whether delivery is to an address in a PO Box only zone, after NCOA <sup>Link</sup> processing.
AFTER_NCOA_POST_DIRECTIONAL	Post-directional of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_PRE_DIRECTIONAL	Predirectional of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_PRIMARY_NUMBER	Primary number of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_QUERY_ZIP_CODE	
AFTER_NCOA_RECORD_TYPE	Record type for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_STATE	State for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_STREET_NAME	Street name of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_SUFFIX	Suffix of the input record after NCOA <sup>Link</sup> processing.

Field ID	Description
AFTER_NCOA_SUITELINK_FOOTNOTE	<p>The Suite<sup>Link</sup> footnote code for the input record after NCOA<sup>Link</sup> processing:</p> <ul style="list-style-type: none"> <li>• "" – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup within the Suite<sup>Link</sup> file. Only default high-rise addresses qualify for a Suite<sup>Link</sup> lookup.</li> <li>• A – The address was processed and secondary information was added to the resulting address.</li> <li>• 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.</li> </ul>
AFTER_NCOA_UNIT_DESIGNATOR	Unit designator of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_UNIT_NUMBER	Unit number of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_URBANIZATION	Urbanization of the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_USPS_4_STATE_BARCODE	The Intelligent Mail barcode for the input record after NCOA <sup>Link</sup> processing.
AFTER_NCOA_ZIP_CODE	ZIP Code of the input record after NCOA <sup>Link</sup> processing.
ANK_FOOTNOTE	
BEFORE_CASS_ADDRESS_LINE_1	Address Line 1 of the input record before address correction.
BEFORE_CASS_ADDRESS_LINE_2	Address Line 2 of the input record before address correction.

Field ID	Description
BEFORE_CASS_BUSINESS	Business name of the input record before address correction.
BEFORE_CASS_CARRIER_ROUTE	Carrier route for the input record before address correction.
BEFORE_CASS_CITY	City of the input record before address correction.
BEFORE_CASS_DPC	DPC for the input record before address correction.
BEFORE_CASS_FIRST_NAME	First name of the input record before address correction.
BEFORE_CASS_FULL_NAME	Full name of the input record before address correction.
BEFORE_CASS_LAST_LINE	City, State, and ZIP Code of the input record before address correction.
BEFORE_CASS_LAST_NAME	Last name of the input record before address correction.
BEFORE_CASS_MIDDLE_NAME	Middle name of the input record before address correction.
BEFORE_CASS_POST_DIRECTIONAL	Post-directional of the input record before address correction.
BEFORE_CASS_PRE_DIRECTIONAL	Pre-directional of the input record before address correction.
BEFORE_CASS_PREFIX_TITLE	Prefix title of the input record before address correction.
BEFORE_CASS_PRIMARY_NUMBER	Primary number of the input record before address correction.

Field ID	Description
BEFORE_CASS_STATE	State of the input record before address correction.
BEFORE_CASS_STREET_NAME	Street name of the input record before address correction.
BEFORE_CASS_SUFFIX	Suffix of the input record before address correction.
BEFORE_CASS_SUFFIX_TITLE	Suffix title of the input record before address correction.
BEFORE_CASS_UNIT_DESIGNATOR	Unit designator of the input record before address correction.
BEFORE_CASS_UNIT_NUMBER	Unit number of the input record before address correction.
BEFORE_CASS_URBANIZATION	Urbanization of the input record before address correction. (Puerto Rico addresses only.)
BEFORE_CASS_ZIP_CODE	ZIP Code of the input record before address correction.
BUSINESS	The name of the business.
CARRIER_ROUTE	The postal carrier route designation for the input record.
CASSDATE	Contains information about when address was last processed.
CITY	City; required for CASS processing unless LAST_LINE specified.
CITY_ABBREVIATED	Abbreviated city of the input record after address correction and before NCOA <sup>Link</sup> processing.
CONGRESSIONAL_DISTRICT	The congressional district number for this address.
COUNTRY	Name of the Country.

Field ID	Description
COUNTY_CODE	5-digit county code for the input record.
COUNTY_NAME	Name of the County.
DP_BARCODE	Delivery Point barcode string for the input record after address correction and before NCOA <sup>Link</sup> processing.
DPC	DPC for the input record after address correction and before NCOA <sup>Link</sup> processing.
DPV_CODED	Indicates if the address was confirmed with DPV.

Field ID	Description
DPV_FOOTNOTE	<p data-bbox="440 302 719 333">DPV footnote code:</p> <ul data-bbox="509 373 1490 1556" style="list-style-type: none"><li data-bbox="509 373 1451 405">• LK - Processing locked out due to a seed record being processed.</li><li data-bbox="509 453 959 485">• AA - Matched to the ZIP+4 file</li><li data-bbox="509 533 1029 564">• A1 - No match against the ZIP+4 file</li><li data-bbox="509 613 1273 644">• BB - Matched to DPV file (all components confirmed)</li><li data-bbox="509 693 1487 764">• CC - Matched only after removing secondary information; they were presented but invalid.</li><li data-bbox="509 812 1490 844">• N1 - Input primary matched, but high-rise missing secondary number.</li><li data-bbox="509 892 943 924">• M1 - Primary number missing.</li><li data-bbox="509 972 938 1003">• M3 - Primary number invalid.</li><li data-bbox="509 1052 1154 1083">• P1 - Input missing PO, RR or HC box number.</li><li data-bbox="509 1131 1395 1163">• P3 - Failed DPV because of invalid PO, RR or HC box number.</li><li data-bbox="509 1211 1122 1243">• RR - Matched CMRA (found in CMRA file).</li><li data-bbox="509 1291 1411 1323">• R1 - Matched CMRA, but secondary number (i.e., PMB) missing.</li><li data-bbox="509 1371 971 1402">• U1 - Matched unique ZIP Code.</li><li data-bbox="509 1451 959 1482">• G1 - Matched general delivery.</li><li data-bbox="509 1530 948 1562">• F1 - Matched military address.</li></ul>

Field ID	Description
DPV_INDICATOR	<p>Returns a single character that indicates the level of address validity with DPV:</p> <ul style="list-style-type: none"> <li>• Y - Both the primary and secondary (if present) validated against the DPV database.</li> <li>• S - The primary address is valid according to DPV, but the secondary is invalid.</li> <li>• D - The primary address is valid according to DPV, but the address is missing secondary information.</li> <li>• N - The primary address is not valid according to DPV.</li> <li>• "" - The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.</li> <li>• X - The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.</li> <li>• E - The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.</li> </ul>
DPV_IS_CMRA	Indicates if address is a commercial mail-receiving agent.
DPV_IS_PBSA	Indicates if address is a Post Office Box street address (PBSA).
DPV_IS_VACANT	Indicates if address is unoccupied.
ERROR_CODE	Error code returned from address correction process.
ERROR_STRING	Text that explains the results from CASS processing.
EWS_CODED	Indicates that this record was flagged by the early warning system as a new address not yet included in the USPS databases.
EXTRA_INFO	Extra information.

Field ID	Description
FIRST_NAME	First name.
FULL_NAME	Full name.
LACS_CODED	Indicates whether this address was flagged by LACS <sup>Link</sup> processing.
LACS_FOOTNOTE	Code indicating specific information about a LACS <sup>Link</sup> lookup: <ul style="list-style-type: none"> <li>• "" - Not processed / Seed record.</li> <li>• 00 - No match.</li> <li>• 09 - Matched to default high-rise address; address not updated.</li> <li>• 14 - Match failure to build new address.</li> <li>• 92 - Match secondary dropped from input.</li> <li>• A - Match success.</li> </ul>
LACS_INDICATOR	A single character indicating the result of a LACS <sup>Link</sup> lookup: <ul style="list-style-type: none"> <li>• "" - Not processed.</li> <li>• N - Match / Matched, but there was a failure to build new address.</li> <li>• Y - Match success.</li> <li>• S - Match with secondary dropped from input.</li> <li>• F - Seed record.</li> </ul>
LAST_LINE	Field containing city, state and ZIP Code; required for CASS processing unless CITY, STATE and ZIPCODE already specified.
LAST_NAME	Last name.
LOT_NUMBER	Line-of-Travel number.

Field ID	Description
MATCH_FLAG	Match flag returned from NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"><li>• M – Matched; updated address.</li><li>• F – Foreign move; new address unavailable.</li><li>• K – No forwarding address; new address unavailable.</li><li>• G – PO box closed; new address unavailable.</li><li>• N – No match.</li><li>• X – Other.</li></ul>
MATCHED_TO_DEFAULT	TRUE if record should contain additional secondary address information (but is not required to).
MIDDLE_NAME	The middle name of the person on this record.
MOVE_EFFECTIVE	The date that the matched move became or becomes the record's active address.

Field ID	Description
MOVE_FOOTNOTE	<p data-bbox="440 302 1192 338">Describes the results of the MOVEService processing.</p> <p data-bbox="440 375 964 411">Match Found – new address returned</p> <ul data-bbox="509 447 1328 638" style="list-style-type: none"> <li data-bbox="509 447 899 483">• A – Input record matched</li> <li data-bbox="509 525 1328 560">• 91 – Secondary number dropped from change of address</li> <li data-bbox="509 602 1252 638">• 92 – Secondary number dropped from input record</li> </ul> <p data-bbox="440 714 1003 749">Match Found – new address unavailable</p> <ul data-bbox="509 785 1243 1213" style="list-style-type: none"> <li data-bbox="509 785 781 821">• 1 – Foreign move</li> <li data-bbox="509 863 886 898">• 2 – Move left no address</li> <li data-bbox="509 940 1003 976">• 3 – PO box closed; no forwarding</li> <li data-bbox="509 1018 1040 1054">• 5 – New 11-digit DPBC is ambiguous</li> <li data-bbox="509 1096 1243 1131">• 14 – New address would not convert to deliverable</li> <li data-bbox="509 1173 1036 1209">• 19 – New address not ZIP + 4 coded</li> </ul> <p data-bbox="440 1289 675 1325">No Match Found</p> <ul data-bbox="509 1360 1195 2100" style="list-style-type: none"> <li data-bbox="509 1360 833 1396">• 00 – No move found</li> <li data-bbox="509 1438 1057 1474">• 4 – Street address missing secondary</li> <li data-bbox="509 1516 1195 1551">• 6 – Conflicting directions, middle name related</li> <li data-bbox="509 1593 1114 1629">• 7 – Conflicting directions, gender related</li> <li data-bbox="509 1671 1000 1707">• 8 – Other conflicting instructions</li> <li data-bbox="509 1749 829 1785">• 9 – High-rise default</li> <li data-bbox="509 1827 867 1862">• 10 – Rural route default</li> <li data-bbox="509 1904 1127 1940">• 11 – Individual, insufficient name for match</li> <li data-bbox="509 1982 932 2018">• 12 – Middle name test failed</li> <li data-bbox="509 2060 854 2100">• 13 – Gender test failed</li> </ul>

Field ID	Description
MOVE_ FOOTNOTE_ LONG_ DESCRIPTION	Returns a longer, more detailed description of the results of the MOVEService processing.
MOVE_ FOOTNOTE_ SHORT_ DESCRIPTION	Returns a short description of the results of the MOVEService processing.
MOVE_TYPE	Move type returned from NCOA <sup>Link</sup> processing: <ul style="list-style-type: none"> <li>• I – Individual</li> <li>• F – Family</li> <li>• B – Business</li> </ul>
NAME_ SALUTATION	A greeting (Mr., Mrs, Ms.) for the name on this record.
NAME_SUFFIX	The suffix (Jr., Sr., etc.), if any, for the person on this record.
RECORD_ID	User field that can contain the input record's index or ID.
RECORD_TYPE	The following record types are currently valid: <ul style="list-style-type: none"> <li>• S – Street record</li> <li>• P – Post office box</li> <li>• R – Rural Route or Highway Contract</li> <li>• H – High-rise, Building or Apartment</li> <li>• F – Firm Record</li> <li>• G – General Delivery</li> </ul>

Field ID	Description
STATE	State; required for CASS processing unless LAST_LINE specified.
SUITE_LINK_ FOOTNOTE	<p>Suite<sup>Link</sup> footnote code:</p> <ul style="list-style-type: none"> <li>• "" – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup within the Suite<sup>Link</sup> file. Only default high-rise addresses qualify for a Suite<sup>Link</sup> lookup.</li> <li>• A – The address was processed and secondary information was added to the resulting address.</li> <li>• 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.</li> </ul>
URBANIZATION	Urbanization number. Puerto Rico addresses only.
USER_DEFINED_1	Custom data to attach to a record.
USER_DEFINED_2	Custom data to attach to a record.
USER_DEFINED_3	Custom data to attach to a record.
USER_DEFINED_4	Custom data to attach to a record.
USER_DEFINED_5	Custom data to attach to a record.
USER_DEFINED_6	Custom data to attach to a record.
USER_DEFINED_7	Custom data to attach to a record.
USER_DEFINED_8	Custom data to attach to a record.
USER_DEFINED_9	Custom data to attach to a record.
USER_DEFINED_10	Custom data to attach to a record.
USER_DEFINED_11	Custom data to attach to a record.

Field ID	Description
USER_DEFINED_12	Custom data to attach to a record.
USER_DEFINED_13	Custom data to attach to a record.
USER_DEFINED_14	Custom data to attach to a record.
USER_DEFINED_15	Custom data to attach to a record.
ZIP_CODE	ZIP Code; required for CASS processing unless LAST_LINE specified.
ZIP4_FOOTNOTE	ZIP4 footnote code.

## The .NET PRESORTAssembly Class for Presorting Mailings

The BCC Architect PresortAssembly object performs postal presorting on a list of addresses. PresortAssembly offers a flexible interface, allowing you to specify the amount of information returned for each address, whether mailing lists are returned in presorted order, the number of records sent and returned at a time, how user interface windows are displayed, etc. In addition, PresortAssembly can be configured to incorporate address correction during the presorting process.

### PRESORTAssembly Overview

The following general procedure should be used to implement the PRESORTAssembly class:

1. Add a reference to the .NET component Satori.MRTK.USAssembly.
2. While not required, you can reduce the amount of typing needed by adding using statements (C#) or Imports statements (Visual Basic) for the Satori.MailRoomToolKit.US and Satori.MailRoomToolKit.US Interfaces namespaces.
3. Create a PRESORTAssembly object.
4. Call PrepareTask.
5. Create a USAddressFieldList object that defines the input fields for each record. Fields are added to the USAddressFieldList object using the USFields.Field enumeration. The minimum amount of address data required is ZIP\_CODE (or LAST\_LINE). Alternatively, the entire address can be passed in at once with the ADDRESS\_BLOCK field.

6. Create a `USAddressFieldList` object that defines the updated output fields for each processed record. Of particular interest is `PRESORT_ID`, which indicates a record's position in the sorted list.
7. Call `ShowPresortWizard`. The Presort Wizard allows you to define the settings for a mailing, which can then be saved as a template for future use. By doing this, you can run sorts without showing the wizard.
8. Call `SetProperty` to set the desired properties not set with the Presort Wizard. The individual properties are specified using the `USProperties.Presort` enumeration. You must set `SETTINGS_FIELD_LIST_IN` and `SETTINGS_FIELD_LIST_OUT` using the `USAddressFieldList` objects that were created in steps 5 and 6, respectively. You should also specify the number of records to be processed at a time with `SETTINGS_INPUT_BLOCK_RECORD_COUNT`.
9. Call `ValidateProperties`.
10. Create a `USAddressRecordBlock` object.
11. Loop through the records in your database, and for each record to be processed:
12. Create a `USAddressRecord` object. A `USAddressFieldList` object is required when creating a `USAddressRecord` object. Use the `USAddressFieldList` object created in step 5.
13. Set the input field values using the `Fields` property of the `USAddressRecord` object.
14. Add the `USAddressRecord` object to the `USAddressRecordBlock` object.
15. Repeat steps a through c until the `USAddressRecordBlock` object contains `SETTINGS_INPUT_BLOCK_RECORD_COUNT` `USAddressRecord` objects.
16. Call `Send`, passing in the `USAddressRecordBlock` object.
17. Repeat steps and until all records have been sent. Call the `Clear` function of the `USAddressRecordBlock` object to reset it before adding a new block of records to it.
18. Call `DoSort`.
19. Call `Retrieve`. A new, updated `USAddressRecordBlock` object containing `SETTINGS_RECORD_COUNT_PER_RECEIVE` records will be returned.

20. Iterate through the `USAddressRecordBlock` object to get each updated `USAddressRecord` object. In turn, iterate through each `USAddressRecord` object to get the updated fields for each record. The fields for each output record will match those specified by the `SETTINGS_FIELD_LIST_OUT` property set earlier.
21. Repeat steps and until all records have been retrieved.
22. Call `PreviewPresortReports` or `PrintPresortReports` to preview or print the presort reports. You can also save the reports as a PDF file using the `SavePresortReportsAsPDF` function.
23. Call `EndTask`.

## PRESORTAssembly Functions

The `PRESORTAssembly` functions are defined below. If you have added the `Satori.MRTK.USAssembly` reference to your .NET project, then you can view all of the available `PRESORTAssembly` members, and their definitions, in the Object Browser and IntelliSense.

### AbortTask

#### Syntax

```
void AbortTask();
```

#### Description

Aborts processing.

#### Parameters

None.

#### Return values

None.

#### Notes

Call this function to end processing prematurely.

### DoSort

#### Syntax

```
void DoSort();
```

### Description

Performs the selected sort on all records sent to the sort engine.

### Parameters

None.

### Return values

None.

### Notes

You should call this function after your final call to `Send`.

Addresses will be certified prior to sorting if the `SETTINGS_CASS_PROCESS_FIRST` property is set to true with the `SetProperty` function.

### See also

[Send](#)

## EndTask

### Syntax

```
void EndTask();
```

### Description

Cleans up and releases a `PRESORTAssembly` object.

### Parameters

None.

### Return values

None.

### Notes

`EndTask` cleans up and releases used resources and should be called when done with the `PRESORTAssembly` object.

## GetProperty

### Syntax

```
string GetProperty(USProperties.CASS cassPropertyId);  
string GetProperty(USProperties.Presort presortPropertyId);
```

### Description

Retrieves the current value of a PRESORTAssembly property.

### Parameters

*cassPropertyId*

Specifies the enum name of the CASS property to get.

*presortPropertyId*

Specifies the enum name of the Presort property to get.

### Return values

The value of the property specified by *cassPropertyId* or *presortPropertyId*.

### Notes

All property values are returned as a string, regardless of the data type passed into SetProperty.

This function is overloaded to accept both CASS and Presort properties.

### See also

See the [CASSAssembly Properties](#) section for a complete list of properties.

See the [PRESORTAssembly Properties](#) section for a complete list of properties.

## GetPropertySummary

### Syntax

```
string GetPropertySummary(USProperties.Presort presortPropertyId);
```

### Description

Returns a string containing a description of a property.

#### Parameters

*presortPropertyId*

Specifies the enum name of the property to get information for.

#### Return values

A description of the property specified by presortPropertyId.

#### Notes

This function can be used to display the description of a property on screen to the user.

#### See also

See the [PresortAssembly Properties](#) section for a complete list of properties.

### PrepareTask

#### Syntax

```
void PrepareTask();
```

#### Description

Initializes and prepares the PRESORTAssembly object.

#### Parameters

None.

#### Return values

None.

#### Notes

PrepareTask should be called only once, after the PRESORTAssembly object is created.

This function must be called before calling any of the other functions or setting any of the properties of PRESORTAssembly. Failing to do so will cause subsequent function calls to fail. The one exception to this rule is setting the SETTINGS\_MAILROOM\_SERVER\_LIST property, which must be defined prior to calling PrepareTask.

### PreviewPresortReports

#### Syntax

```
void PreviewPresortReports();
```

**Description**

Previews all selected Presort reports.

**Parameters**

None.

**Return values**

None.

**Notes**

This function previews the reports selected within the Presort Wizard or whose properties have been set to true with SetProperty.

Setting the REPORT\_PRINT\_ALL\_REPORTS property equal to true will set the properties of all reports to true and hence a call to this function will preview all reports.

The previewed reports will appear together in a single window.

**See also**[PreviewReport](#)

See the [PresortAssembly Properties](#) section for the definition of:

[REPORT\\_PRINT\\_ALL\\_REPORTS](#)

[REPORT\\_PRINT\\_CASS\\_FORM](#)

[REPORT\\_PRINT\\_MANIFEST](#)

[REPORT\\_PRINT\\_POSTAGE\\_STATEMENT](#)

[REPORT\\_PRINT\\_POSTAGE\\_SUMMARY](#)

[REPORT\\_PRINT\\_QUALIFICATION](#)

[REPORT\\_PRINT\\_CONTAINER\\_LABELS](#)

[REPORT\\_PRINT\\_ZIP\\_LISTING](#)

[REPORT\\_PRINT\\_ZONE](#)

## PreviewReport

### Syntax

```
void PreviewReport(USReports.Presort mrtkReportId);
```

### Description

Previews a Presort report.

### Parameters

*mrtkReportId*

The ID of the report to preview.

### Return values

None.

### Notes

Unlike `PreviewPresortReports`, which previews all of the selected Presort reports, this function only displays the specified report.

Each call to this function will create a new preview window for the specified report. Hence, multiple calls to `PreviewReport` will result in multiple preview windows being displayed.

Calling this function will override the selections made within the Presort Wizard for the specified report.

To preview the reports specified in the Presort Wizard, call `PreviewPresortReports`.

### See also

[PreviewPresortReports](#)

See the [PRESORTAssembly Reports](#) section for a complete list of reports.

## PrintPresortReports

### Syntax

```
void PrintPresortReports();
```

### Description

Prints all selected Presort reports.

#### Parameters

None..

#### Return values

None.

#### Notes

This function prints the reports selected within the Presort Wizard or whose properties have been set to true with SetProperty.

Setting the REPORT\_PRINT\_ALL\_REPORTS property equal to true will set the properties of all reports to true and hence a call to this function will print all reports.

If the reports have been set to be previewed or saved as a PDF file in the Presort Wizard, then they will be previewed or saved as a PDF instead of printed when calling this function.

#### See also

#### [PrintReport](#)

See the [PresortAssembly Properties](#) section for the definition of:

[REPORT\\_PRINT\\_ALL\\_REPORTS](#)

[REPORT\\_PRINT\\_CASS\\_FORM](#)

[REPORT\\_PRINT\\_MANIFEST](#)

[REPORT\\_PRINT\\_POSTAGE\\_STATEMENT](#)

[REPORT\\_PRINT\\_POSTAGE\\_SUMMARY](#)

[REPORT\\_PRINT\\_QUALIFICATION](#)

[REPORT\\_PRINT\\_CONTAINER\\_LABELS](#)

[REPORT\\_PRINT\\_ZIP\\_LISTING](#)

[REPORT\\_PRINT\\_ZONE](#)

## PrintReport

### Syntax

```
void PrintReport(USReports.Presort mrtkReportId, string printerName,  
bool showPrintSetupDialog);
```

### Description

Prints a Presort report.

### Parameters

*mrtkReportId*

The ID of the report to print.

*printerName*

The name of the printer you wish to print to.

*showPrintSetupDialog*

Set to true to show the printer setup dialog or false to hide this dialog.

### Return values

None.

### Notes

If *printerName* is set to an empty string then the default printer will be used to print the form.

If *showPrintSetupDialog* is set to true then the value for *printerName* will be ignored.

If the printer is a network printer, the *printerName* parameter must be set to the full name (e.g., `\\Server\HP Laser Jet 5`). To find the full printer name, you could print a test page from the print driver.

Unlike `PrintPresortReports`, which prints all of the selected Presort reports, this function only prints the specified report.

Calling this function will override the selections made within the Presort Wizard for the specified report.

To print the reports specified in the Presort Wizard, call `PrintPresortReports`.

See also

[PrintPresortReports](#)

See the [PRESORTAssembly Reports](#) section for a complete list of reports.

## Retrieve

Syntax

```
USAddressRecordBlock Retrieve();
```

Description

Retrieves the processed records from the sort engine.

Parameters

None.

Return values

A USAddressRecordBlock object containing the updated addresses.

Notes

The number of records in the returned record block is determined by the SETTINGS\_RECORD\_COUNT\_PER\_RECEIVE property.

The fields in each USAddressRecord object in the address block are determined by the SETTINGS\_FIELD\_LIST\_OUT property.

See also

See the [PRESORTAssembly Properties](#) section for the definition of:

[SETTINGS\\_FIELD\\_LIST\\_OUT](#)

[SETTINGS\\_RECORD\\_COUNT\\_PER\\_RECEIVE](#)

## SavePresortReportsAsPDF

Syntax

```
string SavePresortReportsAsPDF(string fileName,  
int fileOption);
```

## Description

Saves all selected Presort reports.

## Parameters

### *filename*

The name, including path, of the file to save.

### *fileOption*

Specifies what to do if the supplied file name already exists:

- 0 – Overwrite if file with file name specified in fileName already exists.
- 1 – Prompt if file with file name specified in fileName already exists.
- 2 – Create new file if file with name specified in fileName already exists; a number will be appended to the file name.
- 3 – Append if file with name specified in fileName already exists (not yet implemented)

## Return values

Returns a string containing the path and name of the saved file.

## Notes

This function saves the reports selected within the Presort Wizard or whose properties have been set to true with SetProperty.

Setting REPORT\_PRINT\_ALL\_REPORTS equal to true will set the properties of all reports to true and hence a call to this function will save all reports.

The reports will all be saved as a single file.

## See also

### [SaveReportAsPDF](#)

See the [PresortAssembly Properties](#) section for the definition of:

### [REPORT\\_PRINT\\_ALL\\_REPORTS](#)

### [REPORT\\_PRINT\\_CASS\\_FORM](#)

[REPORT\\_PRINT\\_MANIFEST](#)

[REPORT\\_PRINT\\_POSTAGE\\_STATEMENT](#)

[REPORT\\_PRINT\\_POSTAGE\\_SUMMARY](#)

[REPORT\\_PRINT\\_QUALIFICATION](#)

[REPORT\\_PRINT\\_CONTAINER\\_LABELS](#)

[REPORT\\_PRINT\\_ZIP\\_LISTING](#)

[REPORT\\_PRINT\\_ZONE](#)

## SaveReportAsPDF

### Syntax

```
string SaveReportAsPDF(USReports.Presort mrtkReportId, string  
fileName, int fileOption);
```

### Description

Saves a report as a PDF file.

### Parameters

*mrtkReportId*

The ID of the report to save.

*filename*

The name, including path, of the file to save.

*fileOption*

Specifies what to do if the supplied file name already exists:

- 0 – Overwrite if file with file name specified in fileName already exists.
- 1 – Prompt if file with file name specified in fileName already exists.
- 2 – Create new file if file with name specified in fileName already exists; a number will be

appended to the file name.

- 3 – Append if file with name specified in fileName already exists (not yet implemented)

#### Return values

Returns a string containing the path and name of the saved file.

#### Notes

Unlike `SavePresortReportsAsPDF`, which saves all of the selected Presort reports, this function only saves the specified report.

#### See also

[SavePresortReportsAsPDF](#)

See the [PRESORTAssembly Reports](#) section for a complete list of reports.

## Send

#### Syntax

```
void Send(USAddressRecordBlock addressBlock);
```

#### Description

Sends an address block to the sort engine.

#### Parameters

*addressBlock*

A `USAddressRecordBlock` object containing the input records to be processed.

#### Return values

None.

#### Notes

You may want to experiment with the number of records in an input block as specified by the `SETTINGS_INPUT_BLOCK_RECORD_COUNT` property. In preliminary tests, we have found the optimal setting to be around 25 - 50.

The fields in each `USAddressRecord` object in the address block are determined by the `SETTINGS_FIELD_LIST_IN` property.

You should call DoSort after you have finished sending all of your records to the sort engine.

See also

### [DoSort](#)

See the [PRESORTAssembly Properties](#) section for the definition of:

[SETTINGS\\_FIELD\\_LIST\\_IN](#)

[SETTINGS\\_RECORD\\_COUNT\\_PER\\_RECEIVE](#)

## SetProperty

Syntax

```
void SetProperty(USProperties.CASS cassPropertyId,  
object val)  
  
void SetProperty(USProperties.Presort presortPropertyId, object val)
```

Description

Sets the value of a PRESORTAssembly property.

Parameters

*cassPropertyId*

Specifies the enum name of the CASS property to set.

*presortPropertyId*

Specifies the enum name of the Presort property to set.

*val*

Specifies the new property value.

Return values

None.

Notes

This function is overloaded to accept both CASS and Presort properties.

See also

[GetProperty](#)

[ShowPresortWizard](#)

See the [CASSAssembly Properties](#) section for a complete list of CASS properties.

See the [PRESORTAssembly Properties](#) section for a complete list of Presort properties.

## ShowPresortWizard

### Syntax

```
void ShowPresortWizard();
```

### Description

Displays the Presort Wizard.

### Parameters

None.

### Return values

None.

### Notes

Call this function if you want to display the Presort Wizard. The Presort Wizard provides a graphical interface that leads a user through the steps necessary to do a mailing.

This function should be called before calling Send.

The wizard needs to be displayed only the first time a sort is performed or set up. Using the wizard, the user can specify their mail sort settings and save those settings as a template. This template can then be accessed with the property `TEMPLATE_NAME_TO_USE`.

The `SETTINGS_SHOW_CASS_FORM_CHECKBOX_IN_WIZARD` property must be set to true if you want to print or preview the CASS report.

The page that allows the user to select the reports to print will only be visible if either `SHOW_SORT_PROGRESS` is set to false or `SETTINGS_HIDE_PROGRESS_AFTER_SORT` is set to true before calling this function.

Properties set through the wizard do not have to be set through SetProperty.

See also

[SetProperty](#)

See the [PRESORTAssembly Properties](#) section for the definition of:

[SETTINGS\\_TEMPLATE\\_NAME\\_TO\\_USE](#)

[SETTINGS\\_SHOW\\_CASS\\_FORM\\_CHECKBOX\\_IN\\_WIZARD](#)

[SETTINGS\\_SHOW\\_SORT\\_PROGRESS](#)

[SETTINGS\\_HIDE\\_PROGRESS\\_AFTER\\_SORT](#)

## ValidateProperties

Syntax

```
void ValidateProperties();
```

Description

Verifies that the PRESORTAssembly object is set up correctly and ready to run.

Parameters

None.

Return values

None.

Notes

This function verifies that the basic requirements of a PRESORTAssembly object have been met.

The input fields are verified depending on the type of sort chosen. For example, if an Enhanced Carrier Route (LOT) sort is chosen for a Standard Mail class mailing, then the LOT (Line of Travel) information would need to be included as one of the input fields.

If you have specified that the mailing list should be CASS processed before sorting, then the requirements for the CASSAssembly will also have to be met.

This function needs to be called after setting property values and before calling Send.

See also

[SetProperty](#)

## PRESORTAssembly Properties

The PRESORTAssembly properties are members of the USProperties.Presort enumeration and are defined below. These enum names are used as arguments of the SetProperty and SetProperty functions. These two functions are overloaded to also accept properties from the USProperties.CASS enumeration. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USProperties.Presort and USProperties.CASS enums in the Object Browser and IntelliSense.

### CONTAINER\_LABEL\_MAILER\_CITY\_STATE

Data type

string

Description

Specifies the mailer's city and state to be printed on the tray/sack labels.

Notes

The default value for this property is an empty string.

### CONTAINER\_LABEL\_MAILER\_COMPANY

Data type

string

Description

Specifies the mailer's company name to be printed on the tray/sack labels.

Notes

The default value for this property is an empty string.

### CONTAINER\_LABEL\_MAILER\_ZIP\_CODE

Data type

string

**Description**

Specifies the mailer's ZIP Code to be printed on the tray/sack labels.

**Notes**

The default value for this property is an empty string.

**CONTAINER\_SACK\_LABEL\_COLUMNS****Data type**

int

**Description**

Specifies the number of columns for sack labels.

**Notes**

The default value for this property is 2.

**CONTAINER\_SACK\_LABEL\_CONTINUOUS****Data type**

bool

**Description**

Specifies whether sack labels are to be printed on continuous paper.

**Notes**

Set to true if labels are to be printed on a dot-matrix printer.

The default value for this property is false.

**CONTAINER\_SACK\_LABEL\_HEIGHT****Data type**

float

**Description**

Specifies the height of a sack label, in inches.

**Notes**

The default value for this property is 1.

## CONTAINER\_SACK\_LABEL\_LEFT\_MARGIN

### Data type

float

### Description

Specifies the left page margin, in inches, to start printing text on sack labels

### Notes

The default value for this property is 0.5.

## CONTAINER\_SACK\_LABEL\_ROWS

### Data type

int

### Description

Specifies the number of rows for sack labels.

### Notes

The default values for this property are `1` and `1` for sheet labels and continuous labels, respectively.

## CONTAINER\_SACK\_LABEL\_TOP\_MARGIN

### Data type

float

### Description

Specifies the top page margin, in inches, to start printing text on sack labels

### Notes

The default values for this property are 0.5 and 0 for sheet labels and continuous labels, respectively.

## CONTAINER\_SACK\_LABEL\_WIDTH

### Data type

float

**Description**

Specifies the width of a sack label, in inches.

**Notes**

The default value for this property is 3..

**CONTAINER\_TRAY\_LABEL\_COLUMNS****Data type**

int

**Description**

Specifies the number of columns for tray labels.

**Notes**

The default value for this property is 2.

**CONTAINER\_TRAY\_LABEL\_CONTINUOUS****Data type**

bool

**Description**

Specifies whether tray labels are to be printed on continuous paper.

**Notes**

Set to true if labels are to be printed on a dot-matrix printer.

The default value for this property is false.

**CONTAINER\_TRAY\_LABEL\_HEIGHT****Data type**

float

**Description**

Specifies the height of a tray label, in inches.

**Notes**

The default value for this property is 2.

## CONTAINER\_TRAY\_LABEL\_LEFT\_MARGIN

### Data type

float

### Description

Specifies the left page margin, in inches, to start printing text on tray labels

### Notes

The default value for this property is 0.5.

## CONTAINER\_TRAY\_LABEL\_ROWS

### Data type

int

### Description

Specifies the number of rows for tray labels.

### Notes

The default values for this property are 5 and 7 for sheet labels and continuous labels, respectively.

## CONTAINER\_TRAY\_LABEL\_TOP\_MARGIN

### Data type

float

### Description

Specifies the top page margin, in inches, to start printing text on tray labels

### Notes

The default values for this property are 0.5 and 0 for sheet labels and continuous labels, respectively.

## CONTAINER\_TRAY\_LABEL\_WIDTH

### Data type

float

**Description**

Specifies the width of a tray label, in inches.

**Notes**

The default value for this property is 3.

**DATA\_SERVICES\_CLIENT\_ID\_LIST****Data type**

string

**Description**

Returns a list of client IDs for a specific provider ID that uses BCC Architect Track N Trace.

**Notes**

The default value is an empty string.

**DATA\_SERVICES\_CLIENT\_USER****Data type**

string

**Description**

The name of the client (if any) associated with the current sort template.

**Notes**

The default value is an empty string.

**DATA\_SERVICES\_PASSWORD****Data type**

string

**Description**

The BCC Software Data Services password for the Provider account used with BCC Software Track N Trace.

**Notes**

The default value is an empty string.

## DATA\_SERVICES\_USER

### Data type

string

### Description

The user name for the Data Services Provider account with BCC Software Track N Trace.

### Notes

The default value is an empty string.

## PALLET\_CREATE\_LOW\_VOLUME\_PALLETS

### Data type

bool

### Description

This property will create low volume pallets. This optional sorting preparation will result in the minimum requirement for trays and sacks being overridden for one pallet per PVDS destination in accordance with USPS customer support ruling PS-327. <https://pe.usps.com/text/csr/PS-327.htm>

### Notes

The default value for is FALSE.

## PAYMENT\_ACCOUNT\_NUMBER

### Data type

string

### Description

Specifies the:

- Mail Anywhere account number

-or-

- EPS account number

Accepts up to 20 alphanumeric characters.

## Notes

The default value for this property is an empty string.

**NOTE** You may have set a CAPS account number if you set up your permit before the CAPS system was retired. Update this account number to your EPS ID.

## PAYMENT\_OPTION

### Data type

string

### Description

A string that specifies the Mail.dat MPA postage payment option:

0 – CPP

1 – PVDS

2 – EPS

3 – Debit

4 – Billing

5 – Other

The default value is CPP for periodicals, and Other for other mailing types. Which option you can specify depends on the permit type that you are using.

**NOTE** Option 2 was changed from CAPS to EPS in August – September 2019 Service Pack 1.

## PERMIT\_AGENT\_ADDR

### Data type

string

### Description

Specifies the mailing agent's address.

## Notes

The default value for this property is an empty string.

## PERMIT\_AGENT\_CITY

### Data type

string

### Description

Specifies the mailing agent's city.

### Notes

The default value for this property is an empty string.

## PERMIT\_AGENT\_CONTACT

### Data type

string

### Description

Specifies the mailing agent's contact name.

### Notes

The default value for this property is an empty string.

## PERMIT\_AGENT\_EMAIL

### Data type

string

### Description

Specifies the mailing agent's e-mail address.

### Notes

The default value for this property is an empty string.

## PERMIT\_AGENT\_NAME

### Data type

string

### Description

Specifies the mailing agent's company name.

## Notes

This property is for the company name of the Agent preparing the mailing. An Agent can prepare a mailing on behalf of an organization.

This information and the associated address data only needs to be set if the company preparing the mailing is different from the permit holder.

The default value for this property is an empty string.

## PERMIT\_AGENT\_PHONE

### Data type

string

### Description

Specifies the mailing agent's phone number.

## Notes

The default value for this property is an empty string.

## PERMIT\_AGENT\_STATE

### Data type

string

### Description

Specifies the mailing agent's state.

## Notes

The default value for this property is an empty string.

## PERMIT\_AGENT\_ZIP

### Data type

string

### Description

Specifies the mailing agent's ZIP Code.

## Notes

The default value for this property is an empty string.

## PERMIT\_HOLDER\_ADDR

### Data type

string

### Description

Specifies the permit holder's address.

### Notes

The default value for this property is an empty string.

## PERMIT\_HOLDER\_CAPS\_CUSTOMER\_ID

### Data type

string

### Description

Deprecated. Specifies the permit holder's CAPS customer ID.

If used, the value in this field is combined with that in the PRESORT\_MAILING\_JOB\_ID field and can be used to identify a mailing on the *PostalOne!* dashboard.

### Notes

The Centralized Account Processing System (CAPS) is an electronic payment system.

The default value for this property is an empty string.

## PERMIT\_HOLDER\_CITY

### Data type

string

### Description

Specifies the permit holder's city.

### Notes

The default value for this property is an empty string.

## PERMIT\_HOLDER\_CONTACT

### Data type

string

### Description

Specifies the permit holder's contact name.

### Notes

The default value for this property is an empty string.

## PERMIT\_HOLDER\_EMAIL

### Data type

string

### Description

Specifies the permit holder's e-mail address.

### Notes

The default value for this property is an empty string.

## PERMIT\_HOLDER\_MAILER\_ID

### Data type

string

### Description

This property no longer has any effect.

### Notes

Use PERMIT\_MAIL\_OWNER\_MAILER\_ID or PERMIT\_MAILING\_AGENT\_MAILER\_ID instead.

## PERMIT\_HOLDER\_NAME

### Data type

string

### Description

Specifies the permit holder's company name.

**Notes**

The default value for this property is an empty string.

**PERMIT\_HOLDER\_NONPROFIT\_AUTH\_NO****Data type**

string

**Description**

Specifies the permit holder's nonprofit authorization number.

**Notes**

The default value for this property is an empty string.

**PERMIT\_HOLDER\_PHONE****Data type**

string

**Description**

Specifies the permit holder's phone number.

**Notes**

The default value for this property is an empty string.

**PERMIT\_HOLDER\_STATE****Data type**

string

**Description**

Specifies the permit holder's state.

**Notes**

The default value for this property is an empty string.

**PERMIT\_HOLDER\_ZIP****Data type**

string

**Description**

Specifies the permit holder's ZIP Code.

**Notes**

The default value for this property is an empty string.

**PERMIT\_MAIL\_OWNER\_CRID****Data type**

string

**Description**

The Customer Registration ID for the owner of this mailing.

**Notes**

The default value for this property is an empty string.

**PERMIT\_MAIL\_OWNER\_MAILER\_ID****Data type**

string

**Description**

Specifies the Mailer ID for the owner of the mailing, regardless of whether they are sorting and sending it.

**Notes**

You can specify multiple Mailer IDs, separated by commas. These additional Mailer IDs will be used if you selected Automatic as the PRESORT\_IM\_SEQUENCING\_METHOD, and the first Mailer ID does not have enough unused sequence numbers to cover all mail pieces in the mailing. All other sequencing methods will only use the first Mailer ID.

This will override any Permit template settings.

**PERMIT\_MAILING\_AGENT\_CRID****Data type**

string

**Description**

The Customer Registration ID for the company performing the mailing, if different from the mail owner.

**Notes**

The default value for this property is an empty string.

**PERMIT\_MAILING\_AGENT\_MAILER\_ID****Data type**

string

**Description**

Specifies the Mailer ID for the mailing agent if this mailing is being sent on behalf of another organization.

**Notes**

You can specify multiple Mailer IDs, separated by commas. These additional Mailer IDs will be used if you selected Automatic as the PRESORT\_IM\_SEQUENCING\_METHOD, and the first Mailer ID does not have enough unused sequence numbers to cover all mail pieces in the mailing. All other sequencing methods will only use the first Mailer ID.

This will override any Permit template settings.

**PERMIT\_NUMBER****Data type**

string

**Description**

Specifies the permit number of the permit holder.

**Notes**

The default value for this property is an empty string.

**PERMIT\_ORG\_ADDR****Data type**

string

**Description**

Specifies the mailing organization's address.

**Notes**

The default value for this property is an empty string.

**PERMIT\_ORG\_CITY****Data type**

string

**Description**

Specifies the mailing organization's city.

**Notes**

The default value for this property is an empty string.

**PERMIT\_ORG\_CONTACT****Data type**

string

**Description**

Specifies the mailing organization's contact name.

**Notes**

The default value for this property is an empty string.

**PERMIT\_ORG\_EMAIL****Data type**

string

**Description**

Specifies the mailing organization's e-mail address.

**Notes**

The default value for this property is an empty string.

## PERMIT\_ORG\_MAILER\_ID

### Data type

string

### Description

This property no longer has any effect.

### Notes

Use PERMIT\_MAIL\_OWNER\_MAILER\_ID or PERMIT\_MAILING\_AGENT\_MAILER\_ID instead.

## PERMIT\_ORG\_NAME

### Data type

string

### Description

Specifies the mailing organization's company name.

### Notes

This property is for the company name of the organization on whose behalf a mailing is being prepared.

This information and the associated address data only needs to be set if the company for whom the mailing is being prepared is different from the permit holder.

The default value for this property is an empty string.

## PERMIT\_ORG\_NONPROFIT\_AUTH\_NO

### Data type

string

### Description

Specifies the nonprofit authorization number of the company on whose behalf the mailing is being prepared.

### Notes

The default value for this property is an empty string.

## PERMIT\_ORG\_PHONE

### Data type

string

### Description

Specifies the mailing organization's phone number.

### Notes

The default value for this property is an empty string.

## PERMIT\_ORG\_STATE

### Data type

string

### Description

Specifies the mailing organization's state.

### Notes

The default value for this property is an empty string.

## PERMIT\_ORG\_ZIP

### Data type

string

### Description

Specifies the mailing organization's ZIP Code.

### Notes

The default value for this property is an empty string.

## PERMIT\_OWNER\_GHOST\_NUMBER

### Data type

string

### Description

If the mailing agent uses a permit other than the mail owners', the mail owner or organization must be identified with a number provided by the same post office that issued the permit used for the mailing. This number is used by USPS to track the mail owner when the mail owner does not have a normal mailing permit.

### Notes

The default value an empty string.

This field is limited to 8 alphanumeric characters.

Not used if the user's role in the mailing is mail owner or if the mail owner's permit is being used.

## PERMIT\_POST\_OFFICE

### Data type

string

### Description

Specifies the post office that issued the permit.

### Notes

The default value for this property is an empty string.

## PERMIT\_SECONDARY\_CITY

### Data Type

String

### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the city of the Post Office issuing the secondary permit.

### Notes

The default value is an empty string.

## PERMIT\_SECONDARY\_ACCOUNT\_NUMBER

### Data Type

String

#### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This property specifies the USPS payment account number or the Mail Anywhere account number, which is used in addition to the permit number.

#### Notes

The default value is an empty string.

#### PERMIT\_SECONDARY\_NUMBER

#### Data Type

String

#### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the USPS permit number for the secondary permit.

#### Notes

The default value is an empty string.

#### PERMIT\_SECONDARY\_STATE

#### Data Type

String

#### Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the State of the Post Office issuing the secondary permit.

#### Notes

The default value is an empty string.

#### PERMIT\_SECONDARY\_ZIP

#### Data Type

String

## Description

When using a meter or precanceled stamp permit imprint, an optional secondary mailing permit can be used to pay for any remaining postage due. This specifies the ZIP Code of the Post Office issuing the secondary permit.

## Notes

The default value is an empty string.

## PRESORT\_ACS\_METHOD

### Data type

int

### Description

Determines which service type code is used in the IM Barcode.

### Notes

Possible values:

- 0 – None. This is the default value.
- 1 – ManualCorrection
- 2 – ASR
- 3 – CSR
- 4 – PeriodicalASRwithASE
- 5 – PeriodicalACS
- 6 – TraditionalASROption1
- 7 – TraditionalCSROption1
- 8 – TraditionalASROption2
- 9 – TraditionalCSROption2

- 10 – TraditionalPeriodicalACS
- 11 – ASROption2
- 12 – RSR
- 13 – TRSR
- 14 – TraditionalRSR
- 15 – TraditionalTRSR
- 16 – CSROption2
- 17 – CSR1SD
- 18 – CSR2SD
- 19 – TraditionalCSROption1SD
- 20 – TraditionalCSROption2SD
- 21 – ASR1BM
- 22 – ASR2BM
- 23 – CSR1BM

For an overview of Service Type Identifiers (STIDs), see the [Service Type Identifier \(STID\) Table](#) on the [USPS PostalPro](#) website.

#### PRESORT\_ALWAYS\_USE\_PERIODICALS\_FSS\_PREP

##### Data type

string

##### Description

When preparing a Periodicals mailing, will always use FSS if available.

##### Notes

The default value for this property is 0.

- 0 = do not automatically use FSS for periodicals
- 1 = automatically use FSS for periodicals

## PRESORT\_ANNUAL\_NONSUBSCRIBER\_THRESHOLD\_EXCEEDED

### Data Type

bool

### Description

Determines whether the in-county rates are applied to periodicals.

### Notes

Applies only to periodicals.

Once a periodical has exceeded the annual 10% threshold of mailing to non-subscribers, they can no longer claim in-county rates for those addresses.

Default value is "false."

Set to "true" to indicate the annual threshold has been exceeded.

## PRESORT\_APPLY\_PARCEL\_SURCHARGE

### Data Type

BOOL

### Description

This property will trigger application of the parcel surcharge for the current mailing, if package is not barcoded or if package weighs less than two ounces.

The property has no effect on classes other than St class. For other classes, it is ignored.

### Notes

The default value is FALSE.

## PRESORT\_APPLY\_BREAK\_MARK\_INDICATOR

### Data type

string

**Description**

Specify where to apply a break mark indicator for bundles, trays, or pallets.

**Notes**

0 = apply break mark on first piece

1 = apply break mark on last piece

The default value for this property is 0.

**PRESORT\_BREAK\_MARK\_IND\_BUNDLE****Data type**

string

**Description**

Specifies the break mark indicator to use for bundle breaks.

**Notes**

The default value for this property is "#."

**PRESORT\_BREAK\_MARK\_IND\_CONTAINER****Data type**

string

**Description**

Specifies the break mark indicator to use for container breaks.

**Notes**

The default value for this property is "##."

**PRESORT\_BREAK\_MARK\_IND\_PALLET****Data type**

string

**Description**

Specifies the break mark indicator to use for pallet breaks.

**Notes**

The default value for this property is "###."

**PRESORT\_CARTON\_PIECE\_MAX****Data Type**

Integer

**Description**

Specifies the maximum number of pieces per carton.

**PRESORT\_CARTON\_WEIGHT\_MAX****Data Type**

Integer

**Description**

Specifies the maximum weight per carton, in pounds.

**PRESORT\_CASS\_PROCESS\_DATE\_AUTOMATION****Data type**

string

**Description**

Specifies the CASS processing date to print on the postage statement for automation rate pieces.

**Notes**

The default value for this property is an empty string.

**PRESORT\_CASS\_PROCESS\_DATE\_ECR****Data type**

string

**Description**

Specifies the CASS processing date to print on the postage statement for Enhanced Carrier Route rate pieces.

## Notes

The default value for this property is an empty string.

**PRESORT\_COMBINE\_RESIDUAL\_PIECES**

## Data type

bool

## Description

Use this property to combine 1oz and 2oz residual pieces for a blended rate.

## Notes

The default value for this property is True.

**PRESORT\_CONFIRM\_TRACKING**

## Data type

int

## Description

Determines whether to enable tracking.

Default tracking mechanism is via IMb Tracing.

For Track N Trace service, you must also set PRESORT\_USE\_EASYTRACK to 1.

## Notes

Limited to First Class Machinable Letters, Flats, and Postcards (Automation). Or, Standard Mail Machinable Letters or Flats (Automation or Carrier Route).

Available property values:

Value	Description
0	No tracking
1	Tracking enabled

**NOTE** Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings. It replaces the EasyTrack service. This property now provides access to Track N Trace.

## PRESORT\_CONTAINER\_LABEL\_LAYOUT

### Data type

int

### Description

Determines whether sack and tray labels will be printed as Legacy or Intelligent Mail labels.

### Notes

Intelligent Mail labels are required to qualify for Full Service discounts and services.

The following table lists the available property values:

Value	Description
0	Legacy (default value)
1	Intelligent Mail

## PRESORT\_CREATE\_COURTESY\_PALLETS

### Data type

bool

This property enables the option to be able to create courtesy pallets, which are a special type of user pallet that can be used to place residual bundles in sacks on a pallet for submission of a mailing under the USPS Seamless Acceptance program.

### Notes

The property applies to the Standard Mail Flats and Bound Printed Matter Flats mail piece types. Pallet placards for these pallets are marked as Mixed-NDC.

This property is similar to the PALLET\_CREATE\_LOW\_VOLUME\_PALLETS property, which is useful when you want to place unpalletized containers on a Mixed-NDC pallet when this pallet is allowed according to USPS regulations for the selected Mail Class and Piece Type.

The default value is TRUE.

## PRESORT\_CREATE\_FSF\_SACKS

### Data type

bool

This property enables the option to be able to create FSF sacks. The property applies to machinable Standard Mail Flats, Periodical Flats, and Bound Printed Matter Flats mail piece types.

#### Notes

The property applies to machinable Standard Mail Flats, Periodical Flats and Bound Printed Matter Flats mail piece types.

The default value is FALSE.

### PRESORT\_CREATE\_ORIGIN\_DESTINATION\_CONTAINERS

#### Data type

bool

#### Description

Property to turn off optional preparation of destination Origin trays/sacks.

#### Notes

If set to FALSE, this property will skip the optional preparation of destination Origin trays/sacks for PVDS (Plant Verified Drop Shipment) destinations.

The default value is TRUE.

### PRESORT\_CUSTOM\_BARCODE\_MAILER\_ID

#### Data type

string

#### Description

Third party mailer ID to use for IMb Tracing™.

#### Notes

Must be a Mailer ID registered with USPS® for IMb Tracing.

### PRESORT\_DDU\_ZIP

#### Data type

string

#### Description

List of Destination Entry facilities for the current mailing, delimited by commas.

## Notes

The format for each facility is the 5- or 9-Digit Zip Code.

Carrier Route data is required for DDU sorting. Density and sequencing requirements still apply for Carrier Route eligibility.

## PRESORT\_DROP\_SHIP\_BMC\_ZIPS

### Data type

string

### Description

Specifies BMC facilities for Plant-Verified Drop Shipment.

## Notes

This property is obsolete. Use PRESORT\_DROP\_SHIP\_NDC\_ZIPS instead.

## PRESORT\_DROP\_SHIP\_NDC\_ZIPS

### Data type

string

### Description

Specifies NDC facilities for Plant-Verified Drop Shipment.

## Notes

This property is a list of NDC facilities for Plant-Verified Drop Shipment, delimited by commas. The format for each facility is the 3- or 5-Digit ZIP Code, optionally followed by a colon and minimum weight, in pounds. If a minimum weight is not supplied, the default minimum weight will be used.

The default value for this property is an empty string.

## PRESORT\_DROP\_SHIP\_SCF\_ZIPS

### Data type

string

### Description

Specifies SCF facilities for Plant-Verified Drop Shipment.

## Notes

This property is a list of SCF facilities for Plant-Verified Drop Shipment, delimited by commas. The format for each facility is the 3-Digit ZIP Code, optionally followed by a colon and minimum weight, in pounds. If a minimum weight is not supplied, the default minimum weight will be used.

The default value for this property is an empty string.

## PRESORT\_EXCEPTIONAL\_DISPATCH\_ZIPS

### Data type

string

### Description

Specifies ZIP Codes to be used for Periodicals exceptional dispatch.

## Notes

This property is a list of 5-digit ZIP Codes, delimited by commas, that will be used for exceptional dispatch.

The default value for this property is an empty string.

## PRESORT\_FACING\_SLIP\_COLUMNS

### Data Type

int

### Description

Determines the number of columns to include in the print layout.

## Notes

The default value is 2.

## PRESORT\_FACING\_SLIP\_CONTINUOUS

### Data Type

bool

### Description

Determines if the facing slips are printed on continuous labels. This usually applies to dot-matrix printers.

**Notes**

Default value is "false."

Set to "true" to print facing slips on continuous labels.

**PRESORT\_FACING\_SLIP\_HEIGHT****Data Type**

FLOAT

**Description**

The height of each facing slip in inches if printing to a sheet.

**Notes**

This property is not set if printing to continuous paper.

The default value is 2, or 0 if printing to continuous paper.

**PRESORT\_FACING\_SLIP\_HORIZ\_PITCH****Data Type**

FLOAT

**Description**

Determines the horizontal distance between the left edge of the leftmost facing slip and the left edge of the facing slip to the right of it on the page.

**Notes****PRESORT\_FACING\_SLIP\_LEFT\_MARGIN****Data Type**

FLOAT

**Description**

The left margin in inches when printing facing slips.

**Notes**

The default value is 0.5 inches.

## PRESORT\_FACING\_SLIP\_PAGE\_HEIGHT

Data Type

FLOAT

Description

Determines the vertical dimension of the facing slip, in inches.

## PRESORT\_FACING\_SLIP\_PAGE\_WIDTH

Data Type

FLOAT

Description

Determines the horizontal dimension of the facing slip, in inches.

## PRESORT\_FACING\_SLIP\_PRINT\_ORDER

Data Type

int

Description

Specifies the order in which to print facing slips on the page.

Notes

The values for specifying print order:

Value	Description
0	Default. Z-order (left to right & top to bottom)
1	N-order (Top to Bottom & left to right)
2	Cut stack order

## PRESORT\_FACING\_SLIP\_ROWS

Data Type

int

**Description**

Determines the number of rows of facing slips to print on each page.

**Notes****PRESORT\_FACING\_SLIP\_TOP\_MARGIN****Data Type**

FLOAT

**Description**

The top margin in inches when printing facing slips.

**Notes**

The default value is 0.5 inches, 0 if printing to continuous paper.

**PRESORT\_FACING\_SLIP\_VERT\_PITCH****Data Type**

FLOAT

**Description**

Determines the vertical distance between the top edge of the first facing slip and the top edge of the facing slip below it on the page, in inches.

**Notes****PRESORT\_FACING\_SLIP\_WIDTH****Data Type**

FLOAT

**Description**

Determines the horizontal dimension of the facing slip, in inches.

**Notes****PRESORT\_FIRM\_BUNDLE\_MIN****Data type**

int

Description

Determines the minimum number of pieces required to make a firm bundle.

Notes

Enable firm bundles with the PRESORT\_USE\_FIRM\_BUNDLES property.

This property requires an add-on to BCC Architect to function.

Firm bundles only apply to Periodicals mailings.

**PRESORT\_FORCE\_WALK\_SEQUENCE\_SATURATION**

Data type

bool

Description

Determines whether enhanced carrier route pieces are assumed to qualify for saturation rates.

Notes

When this property is true, saturation rates are applied to all ECR pieces without verifying density requirements.

The default value for this property is false.

**PRESORT\_IM\_BARCODE\_MAILER\_ID\_CODE**

Data type

int

Description

Determines which Mailer ID to use.

Notes

Use the following values:

Value	Description
0	List owner as specified by PERMIT_MAIL_OWNER_MAILER_ID. Default value.
1	Mailing agent, as specified by PERMIT_MAILING_AGENT_MAILER_ID.

## PRESORT\_IM\_CONTAINER\_SEQUENCE\_LAST

### Data type

int

### Description

A read-only property that contains the last sequence number used for Intelligent Mail barcodes on a tray or sack.

### Notes

If no sequence numbers have been used previously, this property will return 0.

## PRESORT\_IM\_CONTAINER\_SEQUENCE\_START

### Data type

int

### Description

The first sequence number for Intelligent Mail barcodes on trays or sacks when you specify manual sequencing in `PRESORT_IM_SEQUENCING_METHOD`.

### Notes

The default value for this property is 1.

Trays and sacks must be barcoded to qualify for Full Service Intelligent Mail discounts and services.

## PRESORT\_IM\_EINDUCTION

### Data Type

bool

### Description

Indicates whether the Intelligent Mail mailing uses eInduction.

### Notes

The default value for this property is False.

This property can be set only if you have the Palletization and Mail.dat add-ons and have registered the keys. The keys must be current.

Drop shipment is required. If that condition is met, this property can be set for any pallet-level container that qualifies for Full Service Intelligent Mail. Container barcodes (IMcb) are required and must be unique within a 45-day period.

BCC Architect supports eInduction for the following mail classes:

Sort Class	Piece Type
Periodicals	Letters, Flats
Standard Mail	Letters, Flats
Bound Printed Matter (Package Services)	Flats (barcoded)

#### PRESORT\_IM\_EINDUCTION\_ACCEPT\_MISSHIPPED

##### Data Type

bool

##### Description

Indicates whether there is support for misshipped containers in the eInduction mailing.

##### Notes

This property is optional.

#### PRESORT\_IM\_EINDUCTION\_FAST\_SCHEDULER\_ID

##### Data Type

string

##### Description

Specifies the Fast Scheduler ID.

##### Notes

The default value for this property is an empty string. If you want to use FAST appointments, provide your FAST\_SCHEDULER\_ID, and log on to the FAST site to finalize your appointment details.

## PRESORT\_IM\_PALLET\_SEQUENCE\_LAST

### Data type

int

### Description

A read-only property that contains the last sequence number used for Intelligent Mail barcodes on a pallet.

### Notes

If no sequence numbers have been used previously, this property will return 0.

Palletization is an Add-on for BCC Architect.

## PRESORT\_IM\_PALLET\_SEQUENCE\_START

### Data type

int

### Description

The first sequence number for Intelligent Mail barcodes on pallets when you specify manual sequencing in PRESORT\_IM\_SEQUENCING\_METHOD.

### Notes

The default value for this property is 1.

Pallets, if used, must be barcoded to qualify for Full Service Intelligent Mail discounts and services.

Palletization is an add-on for BCC Architect.

## PRESORT\_IM\_PIECE\_SEQUENCE\_LAST

### Data type

int

### Description

A read-only property that contains the last sequence number used for Intelligent Mail barcodes on a mail piece.

### Notes

If no sequence numbers have been used previously, this property will return 0.

## PRESORT\_IM\_PIECE\_SEQUENCE\_START

### Data type

int

### Description

The first sequence number for Intelligent Mail barcodes on mail pieces when you specify manual sequencing in PRESORT\_IM\_SEQUENCING\_METHOD.

### Notes

The default value for this property is 1.

## PRESORT\_IM\_SEQUENCING\_METHOD

### Data type

int

### Description

Determines the sequencing method for the Intelligent Mail barcode on mail pieces, containers and pallets.

### Notes

The following table lists the available property values:

Value	Description
0	A single number will be used for all barcodes. You will not qualify for Full Service discounts and services with this value.
1	Automatic numbering based on previously used values and valid ranges. Default value.
2	Manual numbering, beginning with the value specified in PRESORT_IM_PIECE_SEQUENCE_START.
3	Sequence number will be based on the value in the field, INTELLIGENT_MAIL_PIECE_SEQUENCE_ID.

## PRESORT\_IN\_COUNTY\_ZIPS

### Data type

string

**Description**

Specifies ZIP Codes to be forced to be treated as in-county for a Periodicals mailing.

**Notes**

This property is a list of 5-digit ZIP Codes, delimited by commas, that will be treated as in-county.

The default value for this property is an empty string.

**PRESORT\_JOB\_ID****Data type**

string

**Description**

Specifies the job ID to be printed on the reports.

**Notes****PRESORT\_LEGACY\_DDU\_SUPPORT****Data type**

bool

**Description**

Specifies that drop shipment reports should not be generated for DDU destinations

**Notes**

If TRUE, Carrier Route DDU rates will be applied to qualified pieces but drop shipment reports will not be generated.

If FALSE, when doing drop shipments PS-8125 reports and PS-3602-C reports will be generated for each DDU drop location for which the mailing qualifies.

Drop shipment paperwork will only be created for a particular DDU if the mailing has the minimum number of pieces necessary to create a container or pallet for that DDU.

The Carrier Route field is required for DDU sorting. Density and sequencing requirements still apply for Carrier Route rate eligibility.

The default value is FALSE.

## PRESORT\_LIMITED\_CIRCULATION

### Data type

string

### Description

.Whether the US sort engine should apply the Limited Circulation discount.

### Notes

The default value for this property is an empty string.

## PRESORT\_LIST\_NAME

### Data type

string

### Description

Specifies the name of the list to be sorted.

### Notes

The list name is printed on various reports.

The default value for this property is "Mailing List."

## PRESORT\_MAIL\_CONTENT

### Data type

integer

### Description

Specifies whether the mailpiece content is eligible for Election Mail or Political Mail discounts.

### Notes

Possible values:

- 0 – No discount is applied; the mailpiece content does not qualify as Election Mail or Political Mail.

- 1 – Official Election Mail
- 2 – Political Campaign Mail

Default value is 0.

## PRESORT\_MAILDAT\_CONTACT\_EMAIL

### Data type

string

### Description

Specifies the Mail.dat contact's email address.

### Notes

The correct format is an alphanumeric of at most 60 characters.

The default value for this property is an empty string.

## PRESORT\_MAILDAT\_CONTACT\_NAME

### Data type

string

### Description

Specifies the Mail.dat contact's name.

### Notes

The correct format is an alphanumeric of at most 30 characters.

The default value for this property is an empty string.

## PRESORT\_MAILDAT\_CONTACT\_PHONE

### Data type

string

### Description

Specifies the Mail.dat contact's phone number.

#### Notes

The correct format is a ten-digit number.

The default value for this property is an empty string.

### PRESORT\_MAILDAT\_CREATE\_PDR

#### Data type

bool

#### Description

In Mail.dat, PDR (Piece Detail Records) files are created along with other mail.dat files if this property is TRUE.

#### Notes

The default value is FALSE.

Either PBC or PDR must be selected to prevent an error.

### PRESORT\_MAILDAT\_INFORMED\_CODE

#### Data type

string

#### Description

Unique code for the Informed Delivery campaign.

#### Notes

The USPS uses this to identify the campaign. This code can be associated with more than one mailing, but in that case, the Campaign Title for all mailings must match.

Maximum length: 40

### PRESORT\_MAILDAT\_INFORMED\_END

#### Data type

string

#### Description

Date the Informed Delivery campaign is to end.

## Notes

The date:

- Cannot be in the past.
- Cannot come before the Start Date.
- Must end at least one day after the Start Date.

Leave blank when there is no end date. Do not replace with zeroes.

## PRESORT\_MAILDAT\_INFORMED\_ID

### Data type

string

### Description

Unique ID individual Informed Delivery campaign. Can be the same ID as the value in the Campaign Code property.

## Notes

Can be the same ID as the value in the Campaign Code property.

## PRESORT\_MAILDAT\_INFORMED\_NAME

### Data type

string

### Description

Descriptive name for your mailing campaign. The name is used in the Informed Delivery email message to identify who sent the mailing.

## Notes

The name is used in the Informed Delivery email message to identify who sent the mailing.

## PRESORT\_MAILDAT\_INFORMED\_REPRESENT

### Data type

string

#### Description

URL pointer to an image to replace the default scanned grayscale image of the mail piece.

#### Notes

Required for Flats; Optional for Letters and postcards.

As a best practice, the USPS recommends that the image match the style, look and feel of the actual mail piece, thereby providing a consistent customer experience.

Ideally, the artwork for this image and the Representative image are closely aligned.

File type: .jpg, .jpeg

Maximum dimensions: 500 pixels height; 780 pixels width

Maximum size: 200 KB

Maximum URL length: 250

### PRESORT\_MAILDAT\_INFORMED\_RIDEALONG

#### Data type

string

#### Description

URL pointer to an image that includes a clear call to action or next steps.

#### Notes

Required.

To attract the attention of your customers, the USPS recommends that this image be in color. It also recommends that the image include your company logo.

Ideally, the artwork for this image and the Representative image are closely aligned.

File type: .jpg, .jpeg

Maximum dimensions: 200 pixels height; 300 pixels width

Maximum size: 200 KB

Maximum URL length: 250

## PRESORT\_MAILDAT\_INFORMED\_START

### Data type

string

### Description

Date the Informed Delivery campaign is to begin.

### Notes

The date cannot be in the past. The USPS recommends that a campaign start 3 days before and end 3 days after the date you have targeted for the mailing to reach in-home.

Leave blank when there is no start date. Do not replace with zeroes.

## PRESORT\_MAILDAT\_INFORMED\_TARGETURL

### Data type

string

### Description

URL pointer to a website or portal to provide a customer with more information.

### Notes

Required.

You can provide a unique URL for each campaign to track a campaign's results.

Maximum URL length: 250

## PRESORT\_MAILDAT\_INFORMED\_TITLE

### Data type

string

### Description

Descriptive name for your mailing campaign. This is for informational purposes and is visible only to the mailer in the Informed Delivery mailer portal.

## PRESORT\_MAILDAT\_MAILING\_FACILITY\_ID

### Data type

string

**Description**

Specifies the Mail.dat mailing facility ID.

**Notes**

The correct format is an alphanumeric of at most 10 characters.

The default value for this property is an empty string.

**PRESORT\_MAILDAT\_MAILING\_TITLE****Data type**

string

**Description**

The title of the mailing as used in the Mail.dat file.

**Notes**

The default value for this property is an empty string.

**PRESORT\_MAILDAT\_PERMIT\_HOLDER\_ID****Data type**

string

**Description**

Specifies the Mail.dat permit holder ID.

**Notes**

The correct format is a string of at most 8 characters.

The default value for this property is an empty string.

**PRESORT\_MAILDAT\_USER\_LICENSE\_CODE****Data type**

string

**Description**

Specifies the Mail.dat user license code.

#### Notes

The correct format is a four-character alphanumeric starting with a letter.

The default value for this property is an empty string.

### PRESORT\_MAILDAT\_VERIFICATION\_FACILITY\_NAME

#### Data type

string

#### Description

Specifies the Mail.dat verification facility name.

#### Notes

The correct format is an alphanumeric of at most 30 characters.

The default value for this property is an empty string.

### PRESORT\_MAILDAT\_VERIFICATION\_FACILITY\_ZIP4

#### Data type

string

#### Description

Specifies the Mail.dat verification facility ZIP+4 Code.

#### Notes

The correct format is a nine-digit number.

The default value for this property is an empty string.

### PRESORT\_MAILDAT\_VERSION

#### Data type

string

#### Description

Sets the version of the Mail.dat files that will be generated.

#### Notes

For mailings on or after January 24, 2018, set this to either "17-1" or "18-1".

The default value is "18-1".

**IMPORTANT** Mail.dat versions can change frequently, check with *PostalOne!* and BCC Software Technical Support for more details.

## PRESORT\_MAILER\_ID\_USED

### Data type

string

### Description

A read-only property that contains the Mailer ID that was used when multiple Mailer IDs are specified.

## PRESORT\_MAILING\_DATE

### Data type

string

### Description

Specifies the date of the mailing.

### Notes

The default value for this property is the current date.

## PRESORT\_MAILING\_DROP\_ZIP\_CODE

### Data type

string

### Description

Specifies the ZIP Code where the mailing is deposited.

### Notes

The default value for this property is an empty string.

## PRESORT\_MAILING\_IS\_PLUS\_ONE\_MARRIAGE\_MAIL

### Data type

boolean

#### Description

Indicates if the mailing is Plus One Marriage Mail.

#### Notes

The default value for this property is False.

### PRESORT\_MAILING\_JOB\_ID

#### Data type

string

#### Description

Deprecated. Specifies the job ID to be printed on the reports.

#### Notes

If used, the value in this field is combined with that in the PERMIT\_HOLDER\_CAPS\_CUSTOMER\_ID field and can be used to identify a mailing on the *PostalOne!* dashboard.

### PRESORT\_MAKE\_ONLY\_FULL\_5\_DIGIT\_CARRIER\_ROUTE\_TRAYS

#### Data type

bool

#### Description

Determines whether less-than-full 5-digit carrier route trays are created.

#### Notes

When this property is true, trays of 5-digit carrier route pieces will not be created when less than % full.

The default value for this property is false.

### PRESORT\_MANIFEST\_SEQUENCE\_NUMBER

#### Data Type

String

#### Description

Used to set the Manifest sequence number on the Manifest report.

#### Notes

The default value for this property is an empty string..

### PRESORT\_MAX\_PIECES\_PER\_BUNDLE

#### Data Type

Integer

#### Description

Specifies the maximum number of pieces per bundle.

#### Notes

### PRESORT\_MIN\_POSTAGE\_AMOUNT

#### Data type

int

#### Description

Specifies the minimum postage affixed to each piece for meter minimum and stamp payment types.

#### Notes

The default value for this property is 0.

### PRESORT\_MIXED\_WEIGHT\_SORT

#### Data type

bool

#### Description

Specifies the mailing contains pieces of different weights and thicknesses.

#### Notes

The weight and thickness of each piece should be input for a mixed-weights mailing using the `PIECE_WEIGHT` and `PIECE_THICKNESS` fields, respectively.

The default value for this property is false.

Mixed-weights is a BCC Architect add-on.

## PRESORT\_MOVE\_UPDATE\_DATE

### Data type

String

### Description

Contains the date that this entire mailing list was last processed. This will be printed on the postal forms.

### Notes

You can organize the date in any standard format, i.e., " January ", "01--", "Jan. , ", etc.

The default is an empty string.

## PRESORT\_MOVE\_UPDATE\_METHOD

### Data type

int

### Description

Determines which box is checked in the Move Update Method section of the Postage Statement. Also, determines the Move Update method saved via Mail.dat.

### Notes

The following table lists the available property values:

Value	Description
0	None
1	NCOA <sup>Link</sup>
2	Traditional ACS
3	Ancillary Service Endorsement
4	Alternative
6	Multiple
7	OneCode ACS

Value	Description
8	n/a Alternative Address Format
9	Automatic (If PRESORT_ACS_METHOD is used will select OneCode ACS.)

**IMPORTANT** USPS has retired FastForward. The previously available value "5 – FastForward" is no longer valid as a selection.

### PRESORT\_MULTIPLE\_ENTRY\_POINT

Data Type

String

Description

Not currently implemented.

Notes

The default value for this property is an empty string.

### PRESORT\_ORIGIN\_KEY

Data type

string

Description

Indicates the Locale key for the facility that you want to use for a mailing under the USPS Seamless Acceptance program.

The default value is an empty string.

### PRESORT\_PALLET\_PLACARD\_LAYOUT

Data type

int

Description

Determines the pallet placard format.

Notes

The following table lists the available property values:

Value	Description
0	Legacy format (default value)
1	4 x 7 Intelligent Mail placard
2	8.5 x 11 Intelligent Mail placard

## PRESORT\_PERMIT\_ZIPCODE

### Data Type

String

### Description

The ZIP Code where the permit was issued.

### Notes

The default value for this property is an empty string.

## PRESORT\_PIECE\_WEIGHT

### Data type

string

### Description

Specifies the weight of a mailpiece, in ounces.

### Notes

For mixed-weights mailings, this is the default weight of a piece for which the weight is not provided as input.

The default value for this property is 0.

## PRESORT\_PIECES\_ARE\_BARCODED

### Data type

bool

### Description

Specifies whether the pieces in the mailing bear a barcode.

### Notes

A barcode is required to obtain certain rates, such as automation.

The default value for this property is false.

## PRESORT\_PIECES\_PER\_INCH

### Data type

int

### Description

Specifies the number of mailpieces per inch.

### Notes

The most accurate way of determining this value is to measure a stack of mailpieces and divide the number of pieces by the total thickness of the stack.

For mixed-weights mailings, this is the default value of a piece for which the thickness is not provided as input.

The default value for this property is 0.

## PRESORT\_POSTALONE\_JOB\_ID

### Data type

String

### Description

Returns the PostalOne Job ID, which you can use to search on the PostalOne! portal, provide additional information about the mailer on tags, or append to file names.

### Notes

This property is read-only and can only be retrieved after calling DoSort.

The default is an empty string.

## PRESORT\_PREFERRED\_CONTAINER\_TYPE

### Data type

int

Description

Specifies the preferred container type, if any.

Notes

The following table lists the available property values:

Value	Description
0	No preference
1	1- & 2-foot trays
2	1-foot trays
3	2-foot trays
4	EMM trays
5	Tubs
6	Sacks
7	Carton

**PRESORT\_PRINT\_FACING\_SLIPS**

Data type

bool

Description

Determines whether to print facing slips.

Notes

Default value is "false."

**PRESORT\_PRINT\_PALLET\_LABELS**

Data type

bool

**Description**

Determines whether to print pallet labels.

**Notes**

Default value is "false."

**PRESORT\_PRINT\_TRAY\_LABELS\_ONLY****Data Type**

bool

**Description**

Determines whether to print tray labels only, without additional documents.

**Notes**

Default value is "false."

Set to "true" to print tray labels only, without additional documents.

**PRESORT\_PUB\_INFO\_CONTACT\_NAME****Data type**

string

**Description**

Specifies the contact name for a Periodicals mailing.

**Notes**

The default value for this property is an empty string.

**PRESORT\_PUB\_INFO\_CONTACT\_PHONE****Data type**

string

**Description**

Specifies the contact phone number for a Periodicals mailing.

**Notes**

The default value for this property is an empty string.

## PRESORT\_PUB\_INFO\_EDITION\_ISSUE

### Data type

string

### Description

Specifies either the edition code or issue number for this mailing, both which can be up to six characters.

### Notes

The default value for this property is an empty string.

A volume number plus an issue number constitute an edition code.

You must set this property if you set PRESORT\_PUB\_INFO\_VOLUME\_NUMBER.

## PRESORT\_PUB\_INFO\_ENTRY\_STATE\_ZIP4

### Data type

string

### Description

Specifies the entry city, state and ZIP Code for a Periodicals mailing.

### Notes

The default value for this property is an empty string.

## PRESORT\_PUB\_INFO\_ISSUE\_DATE

### Data type

string

### Description

Specifies the publication issue date for a Periodicals mailing.

### Notes

The default value for this property is an empty string.

## PRESORT\_PUB\_INFO\_ISSUE\_FREQUENCY

### Data type

string

### Description

Specifies the publication issue frequency for a Periodicals mailing.

### Notes

The default value for this property is an empty string.

## PRESORT\_PUB\_INFO\_ISSUE\_FREQUENCY\_PER\_YEAR

### Data type

string

### Description

Specifies the publication issue frequency per year for a Periodicals mailing.

### Notes

The default value for this property is an empty string.

## PRESORT\_PUB\_INFO\_NON\_ADVERTISING\_PERCENT

### Data type

int

### Description

Specifies the percentage of non-advertising content for a Periodicals mailpiece.

### Notes

The default value for this property is 0.

## PRESORT\_PUB\_INFO\_PUB\_NUM

### Data type

string

**Description**

Specifies the publication ID number to be printed on the Qualification Report for a Periodicals mailing.

**Notes**

The default value for this property is an empty string.

**PRESORT\_PUB\_INFO\_RIDE\_ALONG\_WEIGHT****Data type**

float

**Description**

Specifies the ride-along weight for a Periodicals mailing.

**Notes**

The default value for this property is 0.

**PRESORT\_PUB\_INFO\_TITLE****Data type**

string

**Description**

Specifies the title of your publication.

**Notes**

The default value for this property is an empty string.

**PRESORT\_PUB\_INFO\_VOLUME\_NUMBER****Data type**

string

**Description**

Specifies the volume number, which can be up to five characters.

## Notes

You can specify this instead of an edition code. A volume number plus an issue number constitute an edition code.

If you set this property, you must also set `PRESORT_PUB_INFO_EDITION_ISSUE`.

The default value for this property is an empty string.

## `PRESORT_PUB_OWNER_AGENT_NAME`

### Data type

string

### Description

The publication owner name for a Periodicals sort.

## Notes

The default value an empty string.

## `PRESORT_REDUCE_OVERFLOW_CONTAINERS`

### Data type

bool

### Description

Only available for First Class Automation Letters and Postcards and Standard Mail Automation Letters. Set this option to place pieces in full trays of a lower sort level instead of less-than-full overflow trays. This does not affect the cost of these pieces in your mailing; see section 235.6.6 of the DMM for more information.

## Notes

The default value for this property is false.

## `PRESORT_SEQUENCING_DATE_ECR`

### Data type

string

### Description

Used to set the "Date of Carrier Route Sequencing" on the postage statement.

#### Notes

The default value is an empty string.

### PRESORT\_SHIPPING\_SERVICES\_LOGIN\_ID

#### Data type

string

#### Description

In/Out.

The Postal Service Logon ID: the logon ID that you use to sign in the USPS FTP site to which you submit your shipping services files. For more information about submitting files via FTP, see [USPS Publication 199](#).

Alphanumeric; maximum 20 characters.

Required if eVS / Shipping Services manifest files are to be produced.

#### Notes

The default value is an empty string.

### PRESORT\_SHIPPING\_SERVICES\_SEQUENCE\_NUMBER

#### Data type

string

#### Description

The sequence number for shipping services.

Optional. Use this field as a way to differentiate shipping services files if you are submitting multiple files to USPS on the same day.

This alphanumeric field can be up to 4 letters or numbers in length, and is appended to the automatically generated file name.

#### Notes

The default value is an empty string.

### PRESORT\_SIMPLIFIED\_DELIVERY\_STATS\_DATE

#### Data type

string

### Description

Indicates the date that this simplified address mailing list was created.

### Notes

This date will be printed on the postage forms and included in the Mail.dat files and the Mail.XML submission.

You must manually set this date for it to be included in the postal documentation.

## PRESORT\_SORT\_RESULTS\_FLAG

### Data type

int

### Description

Determines how the records are returned after the list has been presorted.

### Notes

The following table lists the available property values:

Value	Description
-1	Let the user choose whether or not to return the list in presorted order.
0	Do not return the records in presorted order (default value).
1	Return the records in presorted order.
2	Return the records in presorted order but do not offer the Label Wizard.

## PRESORT\_STATEMENT\_NUMBER

### Data type

string

### Description

The statement number for this mailing.

### Notes

The default value for this property is an empty string.

## PRESORT\_SUPPRESS\_CARRT\_BASIC\_RATES

### Data type

bool

### Description

Specifies that pieces that do not qualify for Carrier Route Saturation or High Density price should be qualified for Automation instead of Carrier Route Basic price.

### Notes

When sorting for Carrier Route Saturation or High Density price, mail pieces that do not qualify because of carrier route density requirements will go as Automation price instead of Carrier Route Basic price. It is usually less expensive to qualify pieces at the Automation prices when they do not qualify for Saturation or High Density.

The default value for this property is true.

## PRESORT\_SUPPRESS\_IN\_COUNTY\_RATES

### Data type

bool

### Description

Specifies that none of the pieces in a Periodicals mailing qualify for in-county rates.

### Notes

When this property is true, all pieces will be treated as outside-county.

The default value for this property is false.

## PRESORT\_TARE\_WEIGHT\_PALLET

### Data type

string

### Description

Tare weight, in pounds, of the pallet and top cap, if any. Tare weight is the weight of the pallet/cap without any mail pieces, and will be excluded from postage calculations.

### Notes

The default value for this property is an empty string.

## PRESORT\_TARE\_WEIGHT\_PRIMARY\_CONTAINER

### Data type

string

### Description

Tare weight, in pounds, of the primary container. Primary container is generally a sack or one-foot tray in mailings that do not use both one-foot and two-foot trays. Tare weight is the weight of the container without any mail pieces, and will be excluded from postage calculations.

### Notes

The default value for this property is an empty string.

## PRESORT\_TARE\_WEIGHT\_SECONDARY\_CONTAINER

### Data type

string

### Description

Tare weight, in pounds, of the primary container. Secondary container will be used if the mailing uses both one-foot and two-foot trays. Tare weight is the weight of the container without any mail pieces, and will be excluded from postage calculations.

### Notes

The default value for this property is an empty string.

## PRESORT\_TEMPLATE\_LIST

### Data type

string

### Description

Returns a semicolon-delimited string of all currently defined presort templates.

### Notes

This property returns the template list for the current .ini file as specified by the `SETTINGS_INI_FILE_NAME` property.

The default value for this property is an empty string.

## PRESORT\_USE\_ACS\_NONE\_FOR\_SAMPLE\_COPIES

### Data type

boolean

### Description

Indicates if Periodical sample (non-subscriber) copies use the No Corrections service types (STIDs) for Intelligent Mail barcodes (IMb).

### Notes

The default value for this property is **False**.

Set this property to True to apply the Alternative Address Sample Copies No Corrections STIDs to all sample (non-subscriber) copies in a mailing. This means that you are opting not to receive address correction notifications for these pieces, which can result in postage savings. For more information, see [Service Type Identifier \(STID\) Table for Periodicals with Alternative Addressing](#).

**NOTE** When you use the No Corrections STIDs, you will not receive address-related notices via PS Form 3579 or electronic ACS, if the address is vacant or not deliverable. Publications processed by the Postal Automated Redirection System, Computerized Forwarding System, or Remote Forwarding System will be discarded without notice to the publisher

## PRESORT\_USE\_DETACHED\_ADDRESS\_LABEL

### Data type

int

### Description

Specifies what type of Detached Address Label to produce, if any.

### Notes

The following table lists the available property values:

Value	Description
0	No detached address label.
1	Create standard Detached Address Labels.
3	Create Detached Marketing Labels.

## PRESORT\_USE\_EASYTRACK

### Data type

bool

### Description

Specifies that the user has chosen to use Track N Trace for IMb Tracing.

### Notes

Defaults to TRUE if the user has Track N Trace available.

Requires Track N Trace add-on serial number.

**NOTE** Track N Trace® is the BCC Software service for using USPS® Informed Visibility® data and Intelligent Mail barcodes to track mailings. It replaces the EasyTrack service. This property now provides access to Track N Trace.

When Track N Trace is used; sequence numbers and Mailer ID used in the IM Barcode is generated via the service and the Mailer ID and any sequence numbers set by the client are ignored. The client property used to set the sequence method `ptIM_SEQUENCING_METHOD` is also ignored because the Track N Trace server manages sequence numbers when tracking is enabled.

Track N Trace by default uses BCC Architect's Informed Visibility Mail Tracking & Reporting (IV-MTR) compatible Mailer ID, but optionally you can register other mail provider's Mailer IDs by contacting BCC Software and registering. It is recommended that if you register a Mailer ID with BCC Software that will only be used for mailings that are submitted with Track N Trace. Using other Mailer IDs also requires approval via the USPS Informed Visibility office.

<https://gateway.usps.com/eAdmin/view/signin>

Sign in to the Business Customer Gateway portal, and then log on to the IV-MTR service.

Track N Trace requires that an initial BCC Software Data Services account is created. This account creation can be done via the Presort Wizard.

User ID and Password for the Track N Trace Web Portal is obtained during initial BCC Software Data Services account setup or if lost by emailing [BCC Software Technical Support](#).

The Presort Wizard allows the designation of additional fields that can be used for searching on the Track N Trace Web Portal. These include Business name, Contact name, and a Custom Field (`FLD_IM_PIECE_IDENTIFIER`).

Track N Trace reporting and tools can be accessed via the Track N Trace Web Portal: <https://Bc-c.trackntrace.com> ⇨.

## PRESORT\_USE\_EMM\_TRAYS

### Data type

bool

### Description

Specifies that Extended Managed Mail (EMM) trays should be used for the mailing.

### Notes

Set property to true to use EMM trays.

EMM trays must be used for tall letter-size mail that does not fit in a regular tray.

The default value for this property is false.

## PRESORT\_USE\_EXCEPTIONAL\_DISPATCH

### Data type

bool

### Description

Specifies that exceptional dispatch should be used for a Periodicals mailing.

### Notes

Set property to true to use exceptional dispatch.

Exceptional dispatch allows Periodicals mailers to transport pieces directly to destination facilities to improve delivery time.

The default value for this property is false.

## PRESORT\_USE\_FIRM\_BUNDLES

### Data type

bool

### Description

Set this property to TRUE to sort multiple pieces addressed to the same location into a firm bundle.

### Notes

Specify the minimum number of pieces per firm bundle with PRESORT\_FIRM\_BUNDLE\_MIN.

This property requires an add-on to BCC Architect to function.

Firm bundles only apply to Periodicals mailings.

## PRESORT\_USE\_IM\_ONLY

### Data type

bool

### Description

Determines whether BCC Architect uses Intelligent Mail barcodes only.

### Notes

Once you set this property, it will apply to all sorts for all templates.

## PRESORT\_USE\_MAILDAT

### Data type

bool

### Description

Specifies that Mail.dat information should be added to the report XML.

### Notes

Set property to true to generate Mail.dat information.

To actually generate the Mail.dat files, you need to:

1. Set the REPORT\_FOLDER\_NAME\_MAILDAT property.
2. Set REPORT\_SAVE\_MAILDAT to true and call PrintPresortReports or call PrintReport using the USReports.Presort enum member SAVE\_MAILDAT.

The default value for this property is false.

## PRESORT\_USE\_REPOSITIONABLE\_NOTES

### Data type

bool

### Description

Specifies that repositionable notes are attached to the mailpieces.

Notes

USPS® no longer supports repositionable notes. This property no longer supported.

Set property to true if using repositionable notes.

The default value for this property is false.

**PRESORT\_USE\_SIMPLIFIED\_ADDRESSING\_PIECE\_MAX**

Data type

bool

Description

Determines whether to use the suggested maximum number of mail pieces per bundle in a simplified addressing mailing.

Notes

The USPS suggests that you include no more than mail pieces in a bundle.

The default value for this property is true.

**PRESORT\_USE\_USPS\_PROMOTION**

Data type

int

Description

Specifies whether to use a USPS promotion.

Notes

The default value is -1.

Use comma-delimited values to apply multiple promotions, when more than one promotion is available for a given date range.

The following table describes the available promotion values:

Value	Meaning
31	Enable the "Personalized Color Transpromo" promotion, valid from 7/1/2020 – 12/31/2020.

Value	Meaning
32	Enable the "Emerging and Advanced Technology" promotion, valid from 3/1/2020 – 8/31/2020.
33	Enable the "Informed Delivery" promotion, valid from 9/1/2020 – 11/30/2020.
34	Enable the "Tactile, Sensory & Interactive Mailpiece Engagement" promotion, valid from 2/1/2020 – 7/31/2020.
35	Enable the "Mobile Shopping" promotion, valid from 8/1/2020 – 12/31/2020.
-1	No promotion applied. Use this value to unset the promotion if it has been set incorrectly on a template.

### QUERY\_BARCODE\_REQUIRED

#### Data type

bool

#### Description

Indicates if a DP barcode is required on the mailpiece.

#### Notes

This property returns true if a DP barcode is required on the address label for the type of sort performed.

This property is read-only and can only be retrieved after calling DoSort.

The default value for this property is false.

### QUERY\_ENDORSEMENT\_REQUIRED

#### Data type

bool

#### Description

Indicates if an endorsement line is required on the mailpiece.

#### Notes

This property returns true if an endorsement line is required on the address label for the type of sort performed.

This property is read-only and can only be retrieved after calling DoSort.

The default value for this property is false.

### QUERY\_IM\_BARCODE\_REQUIRED

#### Data type

bool

#### Description

Indicates if an Intelligent Mail barcode is required on the mailpiece.

#### Notes

This property returns true if an IMb is required on the address label for the type of sort performed.

This property is read-only and can only be retrieved after calling DoSort.

The default value for this property is false.

### REPORT\_FILE\_NAME\_CONTAINER\_LABELS

#### Data type

string

#### Description

Specifies the path and file name to use when saving the tray/sack labels as a text file.

#### Notes

The default value for this property is an empty string.

### REPORT\_FILE\_NAME\_MAILDAT

#### Data type

String

#### Description

Returns the name of the Mail.dat .zip file or files.

## Notes

This property is read-only and can only be retrieved after calling DoSort.

File names are alphanumeric 8-character strings. The names, for example, L123abcd, are generated as follows: L123 is the name of your IDEAlliance License Code; abcd is a 4-letter code that is randomly generated for jobs run during 20 second increments.

The default is an empty string.

## REPORT\_FOLDER\_NAME\_MAILDAT

### Data type

string

### Description

Specifies the path to the folder in which to store the Mail.dat files.

## Notes

The default value for this property is an empty string.

## REPORT\_MAILDAT\_SAVE\_AS\_ZIP

### Data type

boolean

### Description

Specifies whether to save connected Mail.dat files in a zip file.

## Notes

The default value for this property false; the Mail.dat files are not zipped.

## REPORT\_PRINT\_ALL\_REPORTS

### Data type

bool

### Description

Specifies whether all of the presort reports should be printed.

## Notes

Setting this property to true will set the value of all of the individual report properties (e.g., REPORT\_PRINT\_MAIL\_SORT\_SUMMARY, REPORT\_PRINT\_QUALIFICATION) to true.

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINT\_CASS\_FORM

### Data type

bool

### Description

Determines whether the CASS Summary Report (PS Form ) is printed.

## Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINT\_CONTAINER\_LABELS

### Data type

bool

### Description

Determines whether the container labels are printed.

## Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINT\_CONTAINER\_LISTING

### Data type

bool

#### Description

Determines whether the Container Listing report is printed.

#### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

### REPORT\_PRINT\_DROP\_SHIPMENT

#### Data type

bool

#### Description

Determines whether the Plant-Verified Drop Shipment (PVDS) report (PS Form ) is printed.

#### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

### REPORT\_PRINT\_MAIL\_SORT\_SUMMARY

#### Data type

bool

#### Description

Determines whether the Mail Sort Summary report is printed.

#### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

### REPORT\_PRINT\_MANIFEST

#### Data type

bool

#### Description

Determines whether the Manifest report is printed for mixed-weights mailings.

#### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

### REPORT\_PRINT\_POSTAGE\_STATEMENT

#### Data type

bool

#### Description

Determines whether the Postage Statement is printed.

#### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

### REPORT\_PRINT\_POSTAGE\_SUMMARY

#### Data type

bool

#### Description

Determines whether the Postage Summary report is printed.

#### Notes

The Postage Summary report is only produced for mixed-weights mailings.

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINT\_QUALIFICATION

### Data type

bool

### Description

Determines whether the USPS Qualification Report is printed.

### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINT\_ZIP\_LISTING

### Data type

bool

### Description

Determines whether the ZIP Code Listing report is printed.

### Notes

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINT\_ZONE

### Data type

bool

### Description

Determines whether the Zone Report is printed.

### Notes

The Zone Report is only produced for a Periodicals or Package Services mailing.

The PrintPresortReports, PreviewPresortReports, and SavePresortReportsAsPDF functions will produce all reports whose properties are set to true.

The default value for this property is false.

## REPORT\_PRINTER\_CASS\_FORM

### Data type

string

### Description

Specifies the printer to use for the CASS Summary Report (PS Form ).

### Notes

The default value for this property is an empty string.

## REPORT\_PRINTER\_CONTAINER\_LABELS

### Data type

string

### Description

Specifies the printer to use for the container labels.

### Notes

The default value for this property is an empty string.

## REPORT\_PRINTER\_CONTAINER\_LISTING

### Data type

string

### Description

Specifies the printer to use for the Container Listing report.

### Notes

The default value for this property is an empty string.

## REPORT\_PRINTER\_DEFAULT

### Data type

string

**Description**

Specifies the default printer to use for all reports.

**Notes**

The default value for this property is an empty string.

**REPORT\_PRINTER\_DROP\_SHIPMENT****Data type**

string

**Description**

Specifies the printer to use for the Plant-Verified Drop Shipment (PVDS) report (PS Form ).

**Notes**

The default value for this property is an empty string.

**REPORT\_PRINTER\_FACING\_SLIPS****Data type**

string

**Description**

Specifies the printer to use for the facing slips.

**Notes**

The default value for this property is an empty string.

**REPORT\_PRINTER\_MAIL\_SORT\_SUMMARY****Data type**

string

**Description**

Specifies the printer to use for the Mail Sort Summary report.

**Notes**

The default value for this property is an empty string.

## REPORT\_PRINTER\_MANIFEST

### Data type

string

### Description

Specifies the printer to use for the Manifest report.

### Notes

The Manifest report is only produced for mixed-weights mailings.

The default value for this property is an empty string.

## REPORT\_PRINTER\_PALLET\_LABELS

### Data type

string

### Description

Specifies the printer to use for the pallet labels.

### Notes

The default value for this property is an empty string.

## REPORT\_PRINTER\_POSTAGE\_STATEMENT

### Data type

string

### Description

Specifies the printer to use for the Postage Statement.

### Notes

The default value for this property is an empty string.

## REPORT\_PRINTER\_POSTAGE\_SUMMARY

### Data type

string

**Description**

Specifies the printer to use for the Postage Summary report.

**Notes**

The Postage Summary report is only produced for mixed-weights mailings.

The default value for this property is an empty string.

**REPORT\_PRINTER\_QUALIFICATION****Data type**

string

**Description**

Specifies the printer to use for the USPS Qualification Report.

**Notes**

The default value for this property is an empty string.

**REPORT\_PRINTER\_ZIP\_LISTING****Data type**

string

**Description**

Specifies the printer to use for the ZIP Code Listing report.

**Notes**

The default value for this property is an empty string.

**REPORT\_PRINTER\_ZONE****Data type**

string

**Description**

Specifies the printer to use for the Zone Report.

**Notes**

The Zone Report is only produced for a Periodicals or Package Services mailing.

The default value for this property is an empty string.

## REPORT\_SAVE\_AS\_FILE\_CONTAINER\_LABELS

### Data type

bool

### Description

Determines if the container label files are saved to the specified folder.

### Notes

Set to true to save the container label files.

The default value for this property is false.

## REPORT\_SAVE\_MAILDAT

### Data type

bool

### Description

Determines if the Mail.dat files are saved to the specified folder.

### Notes

Set to true to save the Mail.dat files.

You need to call `PrintPresortReports`, `PreviewPresortReports`, or `SavePresortReportsAsPDF` to actually create the Mail.dat files.

The default value for this property is false.

## REPORT\_SAVE\_SHIPPING\_SERVICES\_FILE

### Data type

bool

### Description

Indicates whether to generate the Shipping Services file.

### Notes

Set to true to save the Shipping Services file.

The default value for this property is false.

## REPORT\_SHIPPING\_SERVICES\_FILE\_PATH

### Data type

string

### Description

The Shipping Services file path and name.

### Notes

The default value for this property is an empty string.

## SETTINGS\_CASS\_PROCESS\_FIRST

### Data type

bool

### Description

Determines if address correction is performed prior to sorting the list.

### Notes

Set to true to perform address correction prior to presorting.

The default value for this property is false.

## SETTINGS\_DATAFILE\_LOCATION

### Data type

string

### Description

Specifies the location of the [Address.cas](#) file.

### Notes

Setting this property will update the [Address.cas](#) path in the [mrtk.ini](#) file.

The default value for this property is an empty string.

## SETTINGS\_FIELD\_LIST\_IN

### Data type

USAddressFieldList

### Description

A USAddressFieldList object that defines which fields are supplied as input to the sort engine.

### Notes

This property defines the input fields that are contained in each USAddressRecord to be processed.

The input field list should contain, at a minimum, either ZIP\_CODE or LAST\_LINE.

## SETTINGS\_FIELD\_LIST\_OUT

### Data type

USAddressFieldList

### Description

A USAddressFieldList object that defines which fields are returned as output from the sort engine.

### Notes

This property defines the output fields that are contained in each USAddressRecord that has been processed.

## SETTINGS\_HIDE\_PROGRESS\_AFTER\_SORT

### Data type

bool

### Description

Determines if the progress dialog box remains visible after processing is complete.

### Notes

Set to true to hide the progress dialog box after processing is complete.

The presort reports can be previewed or printed from the progress dialog box.

The default value for this property is false.

## SETTINGS\_INI\_FILE\_NAME

### Data type

string

### Description

Specifies the full name and path of the .ini file to use.

### Notes

The default value for this property is "MRTK.INI."

Unless specified otherwise, the path is the Windows folder.

## SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT

### Data type

int

### Description

Specifies the number of records contained in an address block.

### Notes

This property specifies the number of USAddressRecord objects contained in a USAddressRecordBlock object.

The size of the record block determines the number of records sent with each call to Send.

## SETTINGS\_MAILDAT\_CREATE\_PBC

### Data Type

bool

### Description

In Mail.dat, PBC (Piece Barcode) files are created along with other Mail.dat files if this property is TRUE.

### Notes

The default value is FALSE.

Either PBC or PDR must be selected to prevent an error.

## SETTINGS\_MAILROOM\_SERVER\_LIST

### Data type

string

### Description

Specifies the location of the BCC Architect Server.

### Notes

Setting this property creates a TCP/IP connection to the BCC Architect Server, which can reside on the local network or virtually anywhere.

This property should be set before calling PrepareTask.

The format is: Server Name (or IP Address):Port.

Currently, going outside of the proxy server might not be supported.

The default value for this property is an empty string.

## SETTINGS\_MRTK\_VERSION

### Data type

string

### Description

Returns the current BCC Architect server version.

### Notes

The default value is an empty string.

## SETTINGS\_PAUSE\_BEFORE\_CONTAINER\_LABELS

### Data type

bool

### Description

Determines if printing is paused before printing container labels.

## Notes

Set to true to pause before printing the tray/sack labels. This allows the user to change the paper in a selected printer before the tray/sack labels are printed.

The default value for this property is true.

## SETTINGS\_PAUSE\_BEFORE\_FACING\_SLIPS

### Data Type

bool

### Description

Determines whether the printer will pause before printing the facing slips.

## Notes

Default value for this property is false.

Set to true to pause before printing the facing slips. This allows the user to change the paper in a selected printer before the facing slips are printed.

## SETTINGS\_PAUSE\_BEFORE\_PALLET\_LABELS

### Data Type

bool

### Description

Determines whether the printer will pause before printing the pallet labels.

## Notes

Default value for this property is true.

Set to true to pause before printing the pallet labels. This allows the user to change the paper stock in the selected printer before the labels are printed.

## SETTINGS\_PREFER\_CONDENSED\_REPORTS

### Data type

bool

### Description

Determines if condensed postal statements are produced when possible.

#### Notes

Set to true to print the EZ version of PS 3600 and PS 3602; false to print the complete version.

The default value for this property is false.

### SETTINGS\_PRINT\_ON\_SERVER

#### Data type

bool

#### Description

Determines where reports are printed when using the BCC Architect Server.

#### Notes

Set to true to print reports on the machine running BCC Architect Server; false to print on the client machine.

The default value for this property is false.

### SETTINGS\_RECORD\_COUNT

#### Data type

int

#### Description

Specifies the total number of records to be processed.

#### Notes

The default value for this property is 0.

### SETTINGS\_RECORD\_COUNT\_PER\_RECEIVE

#### Data type

int

#### Description

Specifies the number of records contained in the address block that is returned by each call to the Retrieve function.

#### Notes

The default value for this property is .

## SETTINGS\_SHOW\_CASS\_FORM\_CHECKBOX\_IN\_WIZARD

### Data type

bool

### Description

Determines if the CASS Report check box is visible in the Presort Wizard and the progress dialog.

### Notes

Set to true to show the CASS report check box; false to hide it.

You may want to set this property to true when SETTINGS\_CASS\_PROCESS\_FIRST is true.

The default value for this property is false.

## SETTINGS\_SHOW\_PRINT\_DIALOG

### Data type

bool

### Description

Determines whether the standard print dialog box is displayed before printing.

### Notes

The default value for this property is false.

## SETTINGS\_SHOW\_SORT\_PROGRESS

### Data type

bool

### Description

Determines if the progress dialog box is displayed during processing.

### Notes

Set to true to display the progress dialog box during processing.

The presort reports can be previewed or printed from the progress dialog box after processing is complete.

The default value for this property is true.

## SETTINGS\_SHOW\_TEMPLATES\_IN\_WIZARD

### Data type

bool

### Description

Determines whether the saved Presort templates are displayed in the Presort Wizard.

### Notes

Set to true to display the saved templates in the Presort Wizard; false to hide them.

The default value for this property is true.

## SETTINGS\_SILENT\_MODE

### Data type

bool

### Description

Determines PRESORTAssembly's mode of operation.

### Notes

Running PRESORTAssembly in silent mode disables all dialogs, including error messages.

The default value for this property is false.

## SETTINGS\_SORT\_WIZARD\_CAPTION

### Data type

string

### Description

Specifies the caption that appears in the title bar of the Presort Wizard.

### Notes

The default value for this property is "Presort Wizard."

## SETTINGS\_TEMPLATE\_NAME\_TO\_USE

### Data type

string

### Description

Specifies the name of the presort template to use for the sort.

### Notes

The default value for this property is an empty string.

## PRESORTAssembly Reports Summary Table

The PRESORTAssembly reports are members of the USReports.Presort enumeration and are defined below. These enum names are used as arguments of the various functions to preview, print or save a report. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USReports.Presort enums in the Object Browser and IntelliSense.

Name	Description
CASS_FORM	CASS Summary Report (PS Form ); this report is required for automation sorts.
CONTAINER_LABELS	Tray/sack labels; this report is required for postal discounts.
CONTAINER_LISTING	A detailed breakdown of all trays/sacks in a mailing.
DROP_SHIPMENT	Plant-Verified Drop Shipment (PVDS) report (PS Form ).
FACING_SLIPS	Specifies whether to print facing slips.
FILE_NAME_CONTAINER_LABELS	Specifies printing the tray/sack labels to a text file.
IM_PIECE_DATA_INPUTS	Stores third-party IMb tracking information. Can include Barcode ID, Service Type ID (STID), Mailer ID (MID), Sequence Number, and/or Routing number used in the mailing.
MAIL_SORT_SUMMARY	Mail Sort Summary Report, a summary of all settings used for the mailing.
MANIFEST	Manifest report for mixed-weights mailings.

Name	Description
PALLET_LABELS	Specifies whether to print pallet labels.
POSTAGE	USPS Postage Statement; the actual form produced depends on the class of the sort; this report is required for postal discounts.
POSTAGE_SUMMARY	Postage Summary Report for mixed-weights mailings.
QUALIFICATION	USPS Qualification Report; this report is required for postal discounts.
SAVE_MAILDAT	The Mail.dat files, which will be saved to the specified folder.
ZIP_LISTING	A detailed breakdown of all ZIP Codes included in the mailing.
ZONE	A detailed breakdown of all zones for Periodicals and Package Services mailings.

### PRESORTAssembly Fields

The PRESORTAssembly fields are members of the USFields.Presort enumeration and are listed below. These enum names are used as arguments of the various functions of the USAddressFieldList and USAddressFields objects. If you have added the Satori.MRTK.USAssembly reference to your .NET project, then you can view all of the available USFields.Presort enums in the Object Browser and IntelliSense.

Field ID	In/out	Data type	Description
ADDRESS_BLOCK			A block of addresses. Sets and retrieves the address as a complete block.

Field ID	In/out	Data type	Description
ADDRESS_LINE_1			The primary address line; required for CASS processing unless ADDRESS_LINE_2 is specified.
ADDRESS_LINE_2			The secondary address line; required for CASS processing unless ADDRESS_LINE_1 is specified.
BREAK_MARK			Returns values for ptBREAK_MARK_IND_BUNDLE, ptBREAK_MARK_IND_CONTAINER, and ptBREAK_MARK_IND_PALLET for records that mark a bundle break, container break, or pallet break. The break marks returned are set by ptAPPLY_BREAK_MARK_INDICATOR
BUNDLE_DEST			The destination of the bundle.
BUNDLE_LEVEL			The level of the bundle.

Field ID	In/out	Data type	Description
BUNDLE_NUMBER			The bundle (formerly referred to as the package) number.
BUSINESS			The name of the business.
CARRIER_ROUTE			The postal carrier route designation for the input record.
CITY			City; required for CASS processing unless LAST_LINE specified.
CONTAINER_DESTINATION			The container destination.
CONTAINER_LEVEL			The container level.
CONTAINER_NUMBER			The container number.
CONTAINER_TYPE			The container type.
CONTAINER_ZIP_CODE			The container's ZIP Code.
COUNTRY			Name of the Country.
COUNTY_CODE			5-digit county code for the input record.

Field ID	In/out	Data type	Description
DP_BARCODE			Delivery Point barcode string for the input record after address correction and before NCOA processing.
DPC			DPC for the input record after address correction and before NCOA processing.
DROP_SITE_KEY	Out	String	Returns the value of the unique key for the drop site for mailpieces that are drop shipped. If the piece is delivered to the origin point, returns "ORIGIN".
ENDORSEMENT_LINE			Endorsement line.
FIRM_BUNDLE_COPIES			The mail piece count to be delivered to this address. Records sorted into a firm bundle will indicate their position in the count, not the total number of copies to be sent to this address.

Field ID	In/out	Data type	Description
FIRM_BUNDLE_INDICATOR			<p>Indicates whether this address is a master record or a duplicated record to account for multiple copies sent to a single address. Is one of the following:</p> <ul style="list-style-type: none"> <li>• C – This is a copied address.</li> <li>• M – This is a master address that has produced copied addresses.</li> <li>• blank – Other address.</li> </ul>
FIRST_NAME			First name.
INTELLIGENT_MAIL_BARCODE			Intelligent Mail barcode (presort field only); requires that the usps4cb.ttf font be installed.

Field ID	In/out	Data type	Description
INTELLIGENT_MAIL_BARCODE_NUMERIC			Human-readable format of the Intelligent Mail barcode (presort field only).
INTELLIGENT_MAIL_CONTAINER_BARCODE			Contains the Intelligent Mail barcode used in the tray or sack that contains this mail piece.
INTELLIGENT_MAIL_PIECE_IDENTIFIER			An optional custom number that can be used by a Track N Trace mailer to identify mail pieces or groups of mail pieces on the Track N Trace Web Portal.
INTELLIGENT_MAIL_PACKAGE_BARCODE			Data needed for Intelligent Mail package barcode (IMpb); requires that the SatIMpb font (72 point) be installed
INTELLIGENT_MAIL_PACKAGE_BARCODE_NUMERIC			Data needed for numeric element below IMpb barcode

Field ID	In/out	Data type	Description
INTELLIGENT_MAIL_ PACKAGE_BARCODE_ HUMAN			Data needed for human readable element above IMpb barcode
INTELLIGENT_MAIL_ PALLET_ BARCODE			Contains the Intelligent Mail barcode used in the pallet that contains this mail piece.
INTELLIGENT_MAIL_ PIECE_ SEQUENCE_ID			Contains the sequence number used in this mail piece's Intelligent Mail barcode. Set this field when you set the property PRESORT_IM_SEQUENCING_METHOD to 3, use input field.
IS_NONSUBSCRIBER			Indicates Periodicals copy is being sent to a non-subscriber
IS_RESIDENCE			Indicates if address is residential. Can be used for Presort processing to qualify Every Door Direct Mail® (EDDM) mailings.

Field ID	In/out	Data type	Description
KEYLINE			Keyline to be printed on address labels for mixed-weight First Class Mail sorts (Manifesting).
LAST_NAME			Last name.
LOT_NUMBER			Line-of-Travel number.
MAILDAT_INFORMED_OPTIN		Bool	Indicates that the mailing is a campaign in the USPS Informed Delivery® program.
MAILDAT_INFORMED_REPRESENT		String	URL pointer to an image to replace the default scanned grayscale image of the mail piece.
MAILDAT_INFORMED_RIDEALONG		String	URL pointer to an image that includes a clear call to action or next steps.
MAILDAT_INFORMED_TARGETURL		String	URL pointer to a website or portal to provide a customer with more information.
PALLET_DESTINATION			Pallet destination.

Field ID	In/out	Data type	Description
PALLET_LEVEL			Pallet level.
PALLET_NUMBER			Pallet number.
PALLET_ZIP_CODE			The pallet's ZIP Code.
PIECE_ENTRY_POINT_DESCRIPTION			The postal point of entry of the piece.
PIECE_HEIGHT			The height of the piece.
PIECE_IN_COUNTY			Indicates if address is in-county. For Periodicals sorts only.
PIECE_LENGTH			The length of the piece.
PIECE_POSTAGE			The exact postage per piece.
PIECE_THICKNESS			Piece thickness; used for mixed-weight First Class Mail sorts (Manifesting).
PIECE_WEIGHT			Piece weight in ounces; used for mixed-weight First Class Mail sorts (Manifesting).

Field ID	In/out	Data type	Description
PIECE_ZONE_NUMBER			Zone number of address for a Periodicals sort.
PRESORT_ID			The rank order of a mail piece in a particular sort.
RECORD_ID			User field that can contain the input record's index or ID.
STATE			State; required for CASS processing unless LAST_LINE specified.
USER_DEFINED_1			Custom data to attach to a record.
USER_DEFINED_2			Custom data to attach to a record.
USER_DEFINED_3			Custom data to attach to a record.
USER_DEFINED_4			Custom data to attach to a record.
USER_DEFINED_5			Custom data to attach to a record.
USER_DEFINED_6			Custom data to attach to a record.
USER_DEFINED_7			Custom data to attach to a record.

Field ID	In/out	Data type	Description
USER_DEFINED_8			Custom data to attach to a record.
USER_DEFINED_9			Custom data to attach to a record.
USER_DEFINED_10			Custom data to attach to a record.
USER_DEFINED_11			Custom data to attach to a record.
USER_DEFINED_12			Custom data to attach to a record.
USER_DEFINED_13			Custom data to attach to a record.
USER_DEFINED_14			Custom data to attach to a record.
USER_DEFINED_15			Custom data to attach to a record.
WALK_SEQUENCE_NUMBER			Walk Sequence number.
ZIP_CODE			ZIP Code; required for CASS processing unless LAST_LINE specified.

# Web Services Reference

## Contents

The Web Services ZIPService Interface for Correcting Single Addresses	790
Processing an Address with ZIPService	791
ZIPService Functions	791
CheckAddress	791
ZIPService Properties	792
AddressInputPreference	792
Address Line Abbreviated	793
ApplyCasingBusiness	793
AssignLOT	793
AssignRDI	794
Casing	794
CityFormat	794
DPVFailureAsError	795
FirmOutput	795
HighwayContractFormat	796
KeepAliasAddress	796
KeepExtraPrimaryData	797
KeepNonMailingCity	797
PMBOutput	797
POBoxFormat	797
PostDirectionalFormat	798
PreDirectionalFormat	798
PrimaryAddressOutput	799
RuralRouteFormat	799
SuffixFormat	800
UnitDesignatorFormat	800
UnitOutput	801
UpdateUncorrectedCityStZip	801
UseDPV	801
UseGeocode	802
UseLACS	802
UseSuiteLink	802
ZIPService Fields	803
ZIPAddress Fields	803
ZIPAddressOutput Fields	807
The Web Services CASSService Interface for Correcting Batch Addresses	819
Using CASSService to Correct Batch Addresses	819
CASSService Functions	820
EndTask	820
GetProperties	821
GetReportsAsPDF	821
PrepareTask	822
SetProperties	822
Update	823
ValidateProperties	824
CASSService Properties	825
CASS_ABBREVIATE_ADDRESS_LINE	825
CASS_CERTIFY_FLAG	825

CASS_DPV_FAILURE_AS_ERROR .....	826
CASS_DUAL_ADDRESS_INPUT_PREFERENCE .....	826
CASS_KEEP_ALIAS_ADDRESS .....	827
CASS_KEEP_EXTRA_PRIMARY_DATA .....	827
CASS_LIST_NAME .....	827
CASS_LIST_PROCESSOR .....	828
CASS_MAILERS_ADDRESS .....	828
CASS_MAILERS_CITY .....	828
CASS_MAILERS_NAME .....	829
CASS_MAILERS_STATE .....	829
CASS_MAILERS_ZIP .....	829
CASS_UPDATE_UNCORRECTED_CITY_ST_ZIP .....	829
CASS_USE_SUITELINK .....	830
FORMAT_CASING .....	830
FORMAT_CITY .....	830
FORMAT_FIRM_OUTPUT .....	831
FORMAT_HIGHWAY_CONTRACT .....	831
FORMAT_PMB_OUTPUT .....	832
FORMAT_PO_BOX .....	832
FORMAT_POST_DIRECTIONAL .....	833
FORMAT_PRE_DIRECTIONAL .....	833
FORMAT_PRIMARY_ADDRESS_OUTPUT .....	833
FORMAT_RURAL_ROUTE .....	834
FORMAT_SUFFIX .....	834
FORMAT_UNIT_DESIGNATOR .....	835
FORMAT_UNIT_OUTPUT .....	835
FORMAT_UPDATE_CASE_BUSINESS .....	836
FORMAT_UPDATE_CASE_NAMES .....	836
SETTINGS_FIELD_LIST_IN .....	836
SETTINGS_FIELD_LIST_OUT .....	837
SETTINGS_INPUT_BLOCK_RECORD_COUNT .....	837
SETTINGS_RECORD_COUNT .....	837
CASSService Fields .....	837
ADDRESS_BLOCK .....	838
ADDRESS_LINE_1 .....	838
ADDRESS_LINE_2 .....	839
BUSINESS .....	839
CARRIER_ROUTE .....	839
CASSDATE .....	839
CITY .....	840
CITY_ABBREVIATED .....	840
CONGRESSIONAL_DISTRICT .....	840
COUNTRY .....	840
COUNTY_CODE .....	840
COUNTY_NAME .....	841
DPC .....	841
DPV_CODED .....	841
DPV_FOOTNOTE .....	841
DPV_INDICATOR .....	842
DPV_IS_CMRA .....	843
DPV_IS_NOSTAT .....	843
DPV_IS_VACANT .....	843
DP_BARCODE .....	843
ERROR_CODE .....	844

ERROR_STRING .....	844
EWS_CODED .....	844
EXTRA_INFO .....	844
FIRST_NAME .....	845
GEOCODE_CENSUS_BLOCK .....	845
GEOCODE_CENSUS_TRACT .....	845
GEOCODE_FOOTNOTE .....	845
GEOCODE_LATITUDE .....	846
GEOCODE_LONGITUDE .....	846
GEOCODE_MSA_CODE .....	846
IS_RESIDENCE .....	847
LACS_CODED .....	847
LACS_FOOTNOTE .....	847
LACS_INDICATOR .....	848
LAST_LINE .....	848
LAST_NAME .....	848
LOT_NUMBER .....	848
MATCHED_TO_DEFAULT .....	849
PMB_NUMBER .....	849
POST_DIRECTIONAL .....	849
PRE_DIRECTIONAL .....	849
RECORD_TYPE .....	849
SKIPPED_CERTIFY .....	850
STATE .....	850
STREET_NAME .....	850
SUFFIX .....	850
SUITELINK_FOOTNOTE .....	851
UNIT_DESIGNATOR .....	851
UNIT_NUMBER .....	851
URBANIZATION .....	851
ZIPCODE .....	852
<a href="#">The Web Services MOVEService Interface for Updating Moved Addresses</a> .....	852
MOVEService Overview .....	853
MOVEService Functions .....	854
DoProcess .....	854
EndTask .....	855
GetProcessStatus .....	855
GetProperties .....	856
GetReportPDF .....	857
PrepareTask .....	857
Retrieve .....	858
Send .....	859
SetProperties .....	859
ValidateProperties .....	860
MOVEService Properties .....	861
CASS_DPV_FAILURE_AS_ERROR .....	861
CASS_DUAL_ADDRESS_INPUT_PREFERENCE .....	861
CASS_KEEP_ALIAS_ADDRESS .....	862
CASS_KEEP_EXTRA_PRIMARY_DATA .....	862
CASS_LIST_NAME .....	862
CASS_LIST_PROCESSOR .....	863
CASS_MAILERS_ADDRESS .....	863
CASS_MAILERS_CITY .....	863
CASS_MAILERS_NAME .....	864

CASS_MAILERS_STATE .....	864
CASS_MAILERS_ZIP .....	864
CASS_UPDATE_UNCORRECTED_CITY_ST_ZIP .....	864
FORMAT_CASING .....	865
FORMAT_CITY .....	865
FORMAT_FIRM_OUTPUT .....	866
FORMAT_HIGHWAY_CONTRACT .....	866
FORMAT_PMB_OUTPUT .....	867
FORMAT_PO_BOX .....	867
FORMAT_POST_DIRECTIONAL .....	867
FORMAT_PRE_DIRECTIONAL .....	868
FORMAT_PRIMARY_ADDRESS_OUTPUT .....	868
FORMAT_RURAL_ROUTE .....	869
FORMAT_SUFFIX .....	869
FORMAT_UNIT_DESIGNATOR .....	870
FORMAT_UNIT_OUTPUT .....	870
FORMAT_UPDATE_CASE_BUSINESS .....	871
FORMAT_UPDATE_CASE_NAMES .....	871
LOGIN_ADMIN_ID .....	872
LOGIN_ADMIN_PASSWORD .....	872
LOGIN_BROKER_ID .....	872
LOGIN_BROKER_PASSWORD .....	873
LOGIN_CUSTOMER_ID .....	873
LOGIN_CUSTOMER_PASSWORD .....	873
MOVE_BUYER_NAME .....	873
MOVE_CLIENT_ID_LIST .....	874
MOVE_CUSTOMER_MAILERID .....	874
MOVE_HIGH_MATCH_RATE_REASON .....	874
MOVE_MAIL_CLASS .....	875
MOVE_MATCH_FLAG .....	876
MOVE_MOVE_MONTH_RANGE .....	876
MOVE_MULTI_NAME_HANDLE .....	877
SETTINGS_FIELD_LIST_IN .....	877
SETTINGS_FIELD_LIST_OUT .....	878
SETTINGS_INPUT_BLOCK_RECORD_COUNT .....	878
SETTINGS_RECORD_COUNT .....	878
MOVEService Fields .....	879
RECORD_ID .....	879
FIRST_NAME .....	880
LAST_NAME .....	880
NAME_SALUTATION .....	880
MIDDLE_NAME .....	881
NAME_SUFFIX .....	881
BEFORE_CASS_FULL_NAME .....	882
BUSINESS .....	882
ADDRESS_LINE_1 .....	882
ADDRESS_LINE_2 .....	882
CITY .....	883
STATE .....	883
ZIP_CODE .....	884
LAST_LINE .....	884
COUNTY_NAME .....	884
COUNTY_CODE .....	885
URBANIZATION .....	885

COUNTRY .....	885
CONGRESSIONAL_DISTRICT .....	886
ADDRESS_BLOCK .....	886
EXTRA_INFO .....	886
CASSDATE .....	886
ERROR_CODE .....	887
ERROR_STRING .....	887
LOT_NUMBER .....	888
CARRIER_ROUTE .....	888
DPC .....	888
DP_BARCODE .....	889
USER_DEFINED fields .....	889
BEFORE_CASS_PRIMARY_NUMBER .....	889
BEFORE_CASS_PRE_DIRECTIONAL .....	890
BEFORE_CASS_POST_DIRECTIONAL .....	890
BEFORE_CASS_STREET_NAME .....	890
BEFORE_CASS_SUFFIX .....	891
BEFORE_CASS_UNIT_NUMBER .....	891
BEFORE_CASS_UNIT_DESIGNATOR .....	891
AFTER_CASS_PMB_NUMBER .....	892
ZIP4_FOOTNOTE .....	892
DPV_CODED .....	892
DPV_IS_CMRA .....	893
DPV_IS_VACANT .....	893
DPV_INDICATOR .....	893
DPV_FOOTNOTE .....	894
LACS_CODED .....	895
EWS_CODED .....	895
RECORD_TYPE .....	896
MATCHED_TO_DEFAULT .....	897
MOVE_EFFECTIVE .....	897
MOVE_TYPE .....	897
MATCH_FLAG .....	898
MOVE_FOOTNOTE .....	898
MOVE_FOOTNOTE_SHORT_DESCRIPTION .....	900
MOVE_FOOTNOTE_LONG_DESCRIPTION .....	900
LACS_FOOTNOTE .....	900
LACS_INDICATOR .....	901
SUITELINK_FOOTNOTE .....	901

## The Web Services ZIPService Interface for Correcting Single Addresses

The ZIPService is a Web Service interface to the ZIPTask library. BCC Software provides this interface as an alternative to the COM and .NET interfaces. This service takes a single address, corrects it using a CASS Certified address correction process and returns it to you.

The ZIPService uses a single function: CheckAddress. This function takes an address, your registration and Add-on key credentials and a properties object. It returns the processed address, along with ZIP + 4, carrier route and barcode information. With an add-on, you can retrieve geocode inform-

ation as well. With just a few lines of code, you can process an address to make it more complete and more likely to arrive at its destination.

## Processing an Address with ZIPService

Use the following procedure to process an address with the ZIPService:

1. Add a Web reference to the ZIPService. The address of the service is <https://ws.sat-orisoftware.com/Architect/us/ZIP/ZIPService.asmx>.
2. Create a ZIPService object.
3. Create a ZIPService.Credentials object and assign the ProductKey and AddOnKeys.
4. Create a ZIPService.ZIPServiceProperties object and set the properties as you wish. We recommend that you set all properties. See the Properties section for more information.
5. Create a ZIPService.ZIPAddress object and build your address with this object.
6. Call CheckAddress on these three objects. Make sure you store the return value using a ZIPAddressOutput object.

## ZIPService Functions

Below are the methods, properties and fields available in ZIPService. If you have added a Web reference to the ZIPService in Visual Studio, you can view all of these functions in the Object Browser.

**NOTE** For the following functions, we use the namespace MyZipServiceReference. This may be different on your system, depending on the development environment. The function example below assumes that the development environment is C#; other environments may use different syntax.

### CheckAddress

#### Syntax

```
MyZipServiceReference.ZipAddressOutput CheckAddress (MyZipServiceReference.Credentials, MyZipServiceReference.ZIPServiceProperties, MyZipServiceReference.ZIPAddress);
```

#### Description

Processes a single address and attempts to correct it.

## Parameters

### *Credentials*

A `MyZipServiceReference.Credentials` object that contains two objects: `ProductKey`, a string that contains your BCC Architect license key, and `AddOnKeys`, an array of strings that contain license keys for your Add-ons.

### *ZIPServiceProperties*

A `MyZipServiceReference.ZIPServiceProperties` object that contains the settings for all properties. We recommend that you set all property values. See below for the list of available properties.

### *ZIPAddress*

A `MyZipServiceReference.ZIPAddress` object that contains the address you want to process. See below for the available fields in this object.

## Return values

A `MyZipServiceReference.ZIPAddressOutput` object that contains your processed address. See below for the valid fields in this object.

## ZIPService Properties

The properties listed below are values within the `ZIPServiceProperties` structure. To set properties for your ZIPService process, instantiate a `ZIPServiceProperties` object, set the member properties listed below and pass this structure to the `CheckAddress()` function.

We recommend that you set all property values. Any property that you do not set will contain the default value, which is usually false, 0 or the first enumeration value. However, this may not always be the case for all programming environments. So, to make sure that you receive the best results for your application, you should set every property.

**NOTE** For the following properties, we use the namespace `MyZipServiceReference`. This may be different on your system, depending on the development environment.

### AddressInputPreference

#### Data type

DualAddressValue (an enumeration)

#### Description

Determines whether to prefer the street address or PO Box for addresses that have both.

## Notes

Use one of the following to set this property:

- `AddressInputPreference.eByPosition` – The bottom address will be used, regardless of which type it is.
- `AddressInputPreference.ePreferPOBox` – Uses PO Box addresses.
- `AddressInputPreference.ePreferStreet` – Uses street addresses.

## Address Line Abbreviated

### Data type

boolean

### Description

Determines if the address lines are returned abbreviated.

## Notes

If you set this to True, the primary address line will be abbreviated to no more than 30 characters.

The default value for this property is False.

## ApplyCasingBusiness

### Data type

boolean

### Description

Determines whether to apply the Casing property to the Business field.

## AssignLOT

### Data type

boolean

### Description

Determines whether to assign the Line-of-Travel number to processed addresses.

## Notes

### AssignRDI

#### Data type

boolean

#### Description

Determines whether to look up the RDI for this address, which indicates if the address is residential or commercial.

## Notes

This property requires an Add-on.

You must purchase a separate data file from the USPS to retrieve this information.

### Casing

#### Data type

CasingValue

#### Description

Determines the case format for the returned address.

## Notes

Use one of the following:

- MyZipServiceReference.CasingValue.eUpper – The address will be returned in all UPPER CASE.
- MyZipServiceReference.CasingValue.eLower – The address will be returned in all lower case.
- MyZipServiceReference.CasingValue.eMixed – The address will be returned in Mixed Case.

### CityFormat

#### Data type

CityFormatValue

### Description

Determines the format of the returned City field.

### Notes

Use one of the following:

- MyZipServiceReference.CityFormatValue.eFullName – Returns the full city name.
- MyZipServiceReference.CityFormatValue.eAbbreviated – Returns the abbreviated name if the city name contains more than 13 characters.
- MyZipServiceReference.CityFormatValue.eAsInput – THIS VALUE IS NO LONGER SUPPORTED.

## DPVFailureAsError

### Data type

boolean

### Description

Determines whether to treat addresses whose secondary address data fails DPV processing as errors.

### Notes

The default for this property is false.

Addresses with missing or invalid secondary information can still be successfully corrected and assigned a ZIP + 4 Code.

## FirmOutput

### Data type

FirmOutputValue

### Description

Determines how firm information is returned for a corrected address.

### Notes

Use one of the following:

- `MyZipServiceReference.FirmOutputValue.eAsInput` – Returns the business name sent as input.
- `MyZipServiceReference.FirmOutputValue.eBusinessNameIfBlank` – Returns firm information only if the business field was blank on input.
- `MyZipServiceReference.FirmOutputValue.eBusinessNameAlways` – Overwrites the contents of the business field with any firm information found.

## HighwayContractFormat

### Data type

`ElementFormatValue`

### Description

Determines how the service will format highway contract information.

### Notes

Set this to one of the following:

- `MyZipServiceReference.ElementFormatValue.eUSPSAbbr` – Abbreviates the information.
- `MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct` – Abbreviates the information using punctuation.
- `MyZipServiceReference.ElementFormatValue.eFullWords` – Outputs the information without abbreviations.

## KeepAliasAddress

### Data type

`boolean`

### Description

Determines whether an input address with a valid street name alias will be allowed or whether they will be replaced with the official USPS street name.

## KeepExtraPrimaryData

### Data type

boolean

### Description

Determines whether to keep any additional and possibly extraneous information in the primary address line.

## KeepNonMailingCity

### Data type

boolean

### Description

Determines if valid but not preferred city names will be returned for corrected addresses.

## PMBOutput

### Data type

PMBOutputValue

### Description

Determines the output location of personal mailbox information, if any.

### Notes

Use one of the following:

- MyZipServiceReference.PMBOutputValue.eWithUnit – PMB information outputs on the same line as the unit information.
- MyZipServiceReference.PMBOutputValue.eAsInput – PMB information outputs the same as in the input address.

## POBoxFormat

### Data type

ElementFormatValue

### Description

Determines how the service will output PO Box information.

### Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

### PostDirectionalFormat

#### Data type

ElementFormatValue

#### Description

Determines how the service will format post-directional information.

#### Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

### PreDirectionalFormat

#### Data type

ElementFormatValue

#### Description

Determines how the service will format pre-directional information.

## Notes

Set this to one of the following:

- `MyZipServiceReference.ElementFormatValue.eUSPSAbbr` – Abbreviates the information.
- `MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct` – Abbreviates the information using punctuation.
- `MyZipServiceReference.ElementFormatValue.eFullWords` – Outputs the information without abbreviations.

## PrimaryAddressOutput

### Data type

`PrimaryAddressOutputValue`

### Description

Determines how the service will output the primary address information.

## Notes

Set this to one of the following:

- `MyZipServiceReference.PrimaryAddressOutputValue.eAddressLine1` – Returns the primary address information as the Address Line 1 field.
- `MyZipServiceReference.PrimaryAddressOutputValue.eAddressLine2` – Returns the primary address information as the Address Line 2 field.

## RuralRouteFormat

### Data type

`ElementFormatValue`

### Description

Determines how the service will format rural route address information.

## Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

## SuffixFormat

### Data type

ElementFormatValue

### Description

Determines how the service will format street suffix information.

### Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

## UnitDesignatorFormat

### Data type

ElementFormatValue

### Description

Determines how the service will format unit type information, such as Suite or Apartment.

### Notes

Set this to one of the following:

- MyZipServiceReference.ElementFormatValue.eUSPSAbbr – Abbreviates the information.
- MyZipServiceReference.ElementFormatValue.eUSPSAbbrPunct – Abbreviates the information using punctuation.
- MyZipServiceReference.ElementFormatValue.eFullWords – Outputs the information without abbreviations.

## UnitOutput

### Data type

string

### Description

Determines on which address line returned unit information is placed.

### Notes

Possible values are:

- ePrimaryAddressLine– Return at the end of the primary address line (default value).
- eSecondaryAddressLineIfBlank– Return on the secondary address line (if secondary address is blank).
- eSecondaryAddressLineAlways– Always return on the secondary address line (existing secondary address data will be overwritten).

## UpdateUncorrectedCityStZip

### Data type

boolean

### Description

Determines if updated City, State and ZIP Code information should be returned for addresses that could not be completely corrected.

## UseDPV

### Data type

boolean

### Description

Determines if addresses should be verified by the DPV process.

### Notes

DPV processing is required for an address to be corrected and assigned a ZIP + 4 Code.

We recommend that you set this property to true.

Without DPV processing, an address will only be confirmed as within a valid range of addresses. DPV processing confirms that the specific street address is a valid delivery point.

### UseGeocode

#### Data type

boolean

#### Description

Determines if the service will return longitude and latitude information for the corrected address.

#### Notes

Geocode requires an add-on from BCC Software.

### UseLACS

#### Data type

boolean

#### Description

Determines if the service will look up LACS<sup>Link</sup> information.

#### Notes

LACS<sup>Link</sup> processing is required for an address to be corrected and assigned a ZIP + 4 Code.

We recommend that you set this property to true.

### UseSuiteLink

#### Data type

boolean

## Description

Determines if the service should use Suite<sup>Link</sup> to locate missing secondary information based on the business field.

## ZIPService Fields

The items below are members of the ZIPAddress and ZIPAddressOutput structures. These items will be divided into two sections, one for each class. The ZIPAddressOutput structure contains all of the fields from the ZIPAddress structure, plus several additional fields.

Assign input address elements to the member objects in a ZIPAddress object before passing that object to the CheckAddress function. The CheckAddress function will return a ZIPAddressOutput structure. This object contains the corrected ZIPAddress fields as well as several other output-only fields.

For an effective ZIPService process, we recommend that you use at least the following fields as input: AddressLine1, City, State and ZipCode.

Note that in all the fields below, we use the namespace MyZipServiceReference. This may be different on your system, depending on the development environment.

## ZIPAddress Fields

### BusinessName

#### Data type

string

#### Description

The name of the business at this address.

#### Notes

This will be used to fill in missing secondary information using Suite<sup>Link</sup> if the UseSuitelink property is set to True.

### AddressBlock

#### Data type

string

#### Description

A complete address block, as would be printed on a mail piece.

## Notes

You can set this field instead of using the AddressLine1, City, State and ZipCode fields.

Each line must be delimited by a carriage return or line feed character.

## AddressLine1

### Data type

string

### Description

The first address line.

## Notes

We recommend that you set this field to run the ZIPService.

## AddressLine2

### Data type

string

### Description

The second address line.

## Notes

This field is not required.

## City

### Data type

string

### Description

The city of this address.

## Notes

We recommend that you set this field to run the ZIPService.

## State

### Data type

string

### Description

The state for this address.

### Notes

We recommend that you set this field to run the ZIPService.

## ZipCode

### Data type

string

### Description

The ZIP Code of this address.

### Notes

We recommend that you set this field to run the ZIPService.

## CityStateZip

### Data type

string

### Description

The City, State and ZIP Code fields formatted as a single line.

### Notes

This is the same information as would print at the bottom of an address block on a mail piece.

## CarrierRoute

### Data type

string

### Description

The carrier route information for this address.

## Street

### Data type

string

### Description

The street name for this address.

### Notes

This information will be created from the primary address information.

## Suffix

### Data type

string

### Description

The street suffix (e.g., St., Ave., Ln.) for this address.

### Notes

This information will be created from the primary address information.

## UnitDesignator

### Data type

string

### Description

The type of unit (e.g., Apt., Ste., Fl.) for the secondary address information.

### Notes

This information will be created from the secondary address information, if any.

## UnitNumber

### Data type

string

### Description

The unit number for this address.

#### Notes

This information will be created from the secondary address information, if any.

#### Urbanization

##### Data type

string

##### Description

The urbanization for this address.

#### Notes

This field applies only to addresses in Puerto Rico.

#### ZIPAddressOutput Fields

#### CarrtCoded

##### Data type

boolean

##### Description

Indicates whether this address was assigned carrier route information.

#### CASSDate

##### Data type

int

##### Description

Encodes the date that this record was last processed.

#### CensusBlock

##### Data type

string

##### Description

The name of census block in which this address is located.

### Notes

A census block is the smallest area for which the census bureau produces information.

You must have the Geocode add-on installed to generate this information.

### CensusTract

#### Data type

string

#### Description

The census tract in which an address is located.

### Notes

A census tract is a geographic area, often coinciding with the limits of cities or towns, defined for census purposes. A census tract can contain several census blocks.

You must have the Geocode add-on installed to generate this information.

### CongressionalDistrict

#### Data type

string

#### Description

The congressional district in which this address is located.

### CountyCode

#### Data type

int

#### Description

The county code for this address.

### CountyFIPSCode

#### Data type

string

### Description

The FIPS (Federal Information Processing Standard) code for the county in which this address is located.

### Notes

This is a 5-digit number. The first two digits represent the state, while the last three digits represent the county.

### CountyName

#### Data type

string

### Description

The name of the county in which this address is located.

### DeliveryPointCheckDigit

#### Data type

string

### Description

Returns the two-digit delivery point and one-digit check digit for the ZIP + 4 Code.

### Notes

The delivery point digits indicate a mail delivery point within a ZIP + 4 Code as to provide a unique number for every deliverable address.

This information is used in POSTNET and Intelligent Mail barcodes.

**NOTE** As of January 27, 2013, POSTNET barcodes do not qualify mailings for Automation discounts.

The check digit verifies that the ZIP + 4 Code is valid and does not contain an error.

### DPBarcodeString

#### Data type

string

### Description

Contains the delivery point barcode as a string.

### Notes

To print this data as a barcode, install the [SatBar.ttf](#) from the BCC Architect disc and apply it to this field in 16 point font.

This is also known as the POSTNET barcode.

**NOTE** As of January 27, 2013, POSTNET barcodes do not qualify mailings for Automation discounts.

### DPVCoded

#### Data type

boolean

#### Description

Indicates that this address was confirmed as a valid delivery point.

### DPVFootnotes

#### Data type

string

#### Description

Indicates the results of the DPV verification process.

### Notes

Returns one or more of the following:

- LK – Processing locked out due to a seed record being processed
- AA – Matched to the ZIP+4 file
- A1 – No match against the ZIP+4 file
- BB – Matched to DPV file (all components confirmed)
- CC – Matched only after removing secondary Information; they were presented but invalid

- N1 – Input Primary matched, but high-rise missing secondary number
- M1 – Primary number missing
- M3 – Primary number invalid
- P1 – Input missing PO, RR, HC box number
- P3 – Failed DPV because of invalid PO, RR, or HC box number
- RR – Matched CMRA (found in CMRA file)
- R1 – Matched CMRA, but secondary number (i.e., PMB) missing
- U1 – Matched unique zip code
- G1 – Matched General delivery
- F1 – Matched military address

#### DPVIndicator

#### Data type

string

#### Description

Contains a single character that describes the result of DPV processing.

#### Notes

Returns one of the following:

- Y – Both the primary and secondary address data validated against the DPV database.
- S – The primary address is valid, but the secondary is invalid.
- D – The primary address is valid, but the address is missing secondary information.
- N – The primary address is not valid. This address was not given a ZIP + 4 Code.
- "" – The address was not presented to the DPV table because it was missing components

needed for the lookup.

- X – The DPV database has been locked-out because of a protocol violation. You must unlock DPV before any more addresses can be processed with DPV.
- E – The DPV data file is more than 105 days old. By USPS restrictions, no more addresses can be presented to the DPV table

#### DPVIsCMRA

##### Data type

boolean

##### Description

Indicates whether this address is a commercial mail-receiving agent.

#### DPVIsNoStat

##### Data type

boolean

##### Description

If True, indicates that the address is in the database, but is not receiving mail yet.

##### Notes

This often indicates the address is new construction, currently being built but not yet occupied and receiving mail.

#### DPVIsVacant

##### Data type

boolean

##### Description

Indicates whether this address has been vacant for at least 90 days.

#### ErrorCode

##### Data type

int

#### Description

A numerical code that indicates the results of CASS processing.

#### Notes

See the appendix for the codes.

#### ErrorCodeString

#### Data type

string

#### Description

Text that explains the results from CASS processing.

#### Notes

This text corresponds to the ErrorCode field.

#### ExtraInfo

#### Data type

string

#### Description

Contains extra address information that the service was unable to parse into any other fields.

#### GeocodeFootnote

#### Data type

string

#### Description

Indicates the results of Geocode processing.

#### Notes

Requires the Geocode add-on.

Contains one of the following:

- 00 – Unable to perform a geocode lookup
- 03 – Geocode data based on a 3-digit ZIP Code
- 05 – Geocode data based on a 5-digit ZIP Code
- 07 – Geocode data based on a 7-digit ZIP Code
- 09 – Geocode data based on a 9-digit ZIP Code

### IsResidence

#### Data type

boolean

#### Description

Indicates whether this address is residential or not.

#### Notes

To use this property, you must have the RDI add-on and subscribe to the RDI service from the USPS.

### LACS

#### Data type

boolean

#### Description

Indicates whether this address was flagged by LACS<sup>Link</sup> processing.

### LACSFootnote

#### Data type

string

#### Description

Provides additional information about the LACS process.

#### Notes

Contains one of the following:

- blank – Not processed / Seed record.
- 00 – No match.
- 09 – Matched to default high-rise address; address not updated.
- 14 – Match failed to build new address.
- 92 – Match secondary dropped from input.
- A – Match success.

### LACSIndicator

#### Data type

string

#### Description

Indicates the results of a LACS process.

#### Notes

Returns one of the following:

- blank – Not processed.
- N – Match, but there was a failure to build new address.
- Y – Match success, new address provided.
- S – Match with secondary dropped from input.
- F – Seed record.

### Latitude

#### Data type

string

#### Description

The latitude of this address.

#### Notes

You must have the Geocode add-on installed to generate this information.

#### Longitude

##### Data type

string

##### Description

The longitude of this address.

#### Notes

You must have the Geocode add-on installed to generate this information.

#### LOTNumber

##### Data type

string

##### Description

The Line-of-Travel number for this address.

#### Notes

Carrier route sorts require this information.

To generate this data, you must have set the assignLOT property to True.

#### MatchedToDefault

##### Data type

boolean

##### Description

Indicates if this address matched to a default ZIP Code.

#### Notes

Default ZIP Codes are for large, single addresses, such as universities or corporations, that do not have +4 codes.

**MSACode**

## Data type

string

## Description

Contains the Metropolitan Statistical Area code for this address.

## Notes

You must have the Geocode add-on installed to generate this information.

**Plus4Coded**

## Data type

boolean

## Description

Indicates whether this address received a valid ZIP + 4 Code.

**PMB**

## Data type

string

## Description

The private mail box number for this address, if any.

## Notes

The default value for this property is an empty string.

**PostDirectional**

## Data type

string

## Description

Contains the post-directional indicator of this address, if any.

## PreDirectional

### Data type

string

### Description

Contains the pre-directional indicator of this address, if any.

## RecordType

### Data type

string

### Description

Indicates the type of address that this is.

### Notes

Contains one of the following characters:

- S – Street record
- P – Post office box
- R – Rural Route or Highway Contract
- H – High-rise, Building or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

## SuiteLinkFootnote

### Data type

string

### Description

Indicates the results of Suite<sup>Link</sup> matching.

## Notes

To generate this data, you must set the use Suite<sup>Link</sup> property to True.

Contains one of the following:

- blank – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup. Only default high rise addresses qualify for a Suite<sup>Link</sup> lookup.
- A – The address was processed and secondary information was added to the resulting address.
- 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.

## The Web Services CASService Interface for Correcting Batch Addresses

The CASService is a Web Service interface to the CASSTask library. BCC Software provides this interface as an alternative to the COM and .NET interfaces. This service takes a group of addresses, processes them using a CASS Certified address correction process and returns the updated addresses to you.

With the CASService, you can process an entire mailing list and prepare it for automation and pre-sorted discounts. All addresses, if validated, will be returned with the correct address information, as well as accurate ZIP + 4 Codes and POSTNET barcodes. When the CASS validation completes, you can then retrieve an address correction report that the USPS will accept as valid documentation of address correction. You can configure the information returned using the SetProperty function. With the Geocode add-on, you can return latitude and longitude information as well.

**NOTE** As of January 27, 2013, POSTNET barcodes do not qualify mailings for Automation discounts.

With just a few lines of code, you can process a list of addresses to make them more complete, more likely to arrive at their destinations and eligible for discounts.

### Using CASService to Correct Batch Addresses

Use the following procedure to process an address with the CASService:

1. Add a Web reference to the CASService. The address of the service is <https://ws.sat-orisoftware.com/Architect/US/CASS/CASSService.asmx>.
2. Create a CASService object.

3. Create a `CASService.Credentials` object and assign the `ProductKey` and `AddOnKeys`.
4. Call `CASService.PrepareTask(MyCredentialsObj)` to verify your registration keys.
5. Create a `CASService.CASServiceProperties` object and set the properties as you wish. We recommend that you set all properties. See the Properties section for more information.
6. Call `CASService.SetProperties(MyPropertiesObj)` to pass your properties to the server.
7. Call `CASService.ValidateProperties()` to verify that all properties are correct.
8. Create a `CASService.AddressBlock` object and build your addresses with this object. The addresses in this object should all have the fields specified by the `SETTINGS_FIELD_LIST_IN` property. The block must have no more than `SETTINGS_INPUT_BLOCK_RECORD_COUNT` number of records. Create as many `AddressBlock` objects as you need. The records in all address blocks must total `SETTINGS_RECORD_COUNT`.
9. Call `CASService.Update` on each `AddressBlock` object. This function will return these addresses as a block, with each record having the fields specified by the `SETTINGS_FIELD_LIST_OUT` property.
10. Save the address correction report with `CASService.GetReportsAsPDF`.
11. Call `CASService.EndTask` to free up all resources used on the server.

## CASService Functions

Below are the methods available in `CASService`. If you have added a Web reference to the `CASService` in Visual Studio, you can view all of these functions in the Object Browser.

**NOTE** For the following functions, we use the namespace `MyCASServiceReference`. This may be different on your system, depending on the development environment. The function syntax assumes that the development environment is `C#`; other environments may use different syntax.

### EndTask

#### Syntax

```
CASService.EndTask();
```

#### Description

Cleans up any server resources and removes all user data from the server.

#### Parameters

None.

#### Return values

None.

#### Notes

EndTask will remove all of your address data from our servers to protect your privacy.

You must call EndTask after you have called Update on your address blocks and retrieved the address correction report PDF.

### GetProperties

#### Syntax

```
PropertyValues MyCASSServiceReference.GetProperties(PropertyIDs);
```

#### Description

Retrieves the values of the specified properties.

#### Parameters

*PropertyIDs*

A list of the properties for which you want to retrieve the values.

#### Return values

GetPropertyValues – A list of the values of the properties specified in PropertyIDs.

#### Notes

#### See also

See the [CASSService Properties](#) section for the names and purposes of the CASSService properties.

### GetReportsAsPDF

#### Syntax

```
PDFFile MyCASSServiceReference.GetReportsAsPDF();
```

#### Description

Retrieves the address correction report as a formatted PDF file.

### Parameters

None.

### Return values

PDFFile – A binary stream that you can save as a PDF file.

### Notes

The CASS Summary Report (PS Form 3553) is the only CASS report currently available.

## PrepareTask

### Syntax

```
MyCASSServiceReference.PrepareTask(  
MyCASSServiceReference.Credentials);
```

### Description

Validates your product and add-on keys and readies the server to accept additional function calls from your client application.

### Parameters

#### *Credentials*

A MyCASSServiceReference.Credentials object that contains two objects: ProductKey, a string that contains your BCC Architect license key, and AddOnKeys, an array of strings that contain license keys for your Add-ons.

### Return values

None.

### Notes

PrepareTask should be called only once, after you create the CASSService object.

You must call this function before calling any of the other functions or setting any of the properties of CASSService. Failing to do so will cause subsequent function calls to fail.

## SetProperties

### Syntax

```
MyCASSServiceReference.SetProperties(Properties);
```

## Description

Sets the values of the specified properties.

## Parameters

### *Properties*

An array of property and value pairs.

## Return values

None.

## Notes

We recommend that you set all properties.

## See also

See the [CASService Properties](#) section for the enum names and purposes of the CASService properties.

## Update

### Syntax

```
CorrectedAddressBlock MyCASServiceReference.Update(UncorrectedAddressBlock);
```

### Description

Corrects the addresses contained in the AddressBlock object.

### Parameters

#### *UncorrectedAddressBlock*

A block of uncorrected addresses. Each address should have the fields listed in SETTINGS\_FIELD\_LIST\_IN.

### Return values

CorrectedAddressBlock – The input addresses, corrected and updated. Each address will have the fields specified in SETTINGS\_FIELD\_LIST\_OUT.

### Notes

Call this function after ValidateProperties. You should call this function for each block of addresses in your mailing list.

You may want to experiment with the `SETTINGS_INPUT_BLOCK_RECORD_COUNT` property. In tests, we have found the optimal setting to be around 25-50. The number of fields that you want returned (see `SETTINGS_FIELD_LIST_OUT` property) greatly affects this number.

For optimal performance, only ask for the output fields that you need. Extra information requires additional lookups that slow processing.

See also

See the [CASSAssembly Properties](#) section for the definition of:

- [SETTINGS\\_FIELD\\_LIST\\_IN](#)
- [SETTINGS\\_FIELD\\_LIST\\_OUT](#)
- [SETTINGS\\_INPUT\\_BLOCK\\_RECORD\\_COUNT](#)

## ValidateProperties

### Syntax

```
Validated MyCASSServiceReference.ValidateProperties();
```

### Description

Validates that the settings will work as configured.

### Parameters

None.

### Return values

Nothing.

### Notes

This function verifies that the basic requirements of a `CASSService` object have been met:

The address matching engine is loaded and able to run.

The input field list consists of the minimum set of fields, namely, `ADDRESS_LINE_1` and either `CITY` and `STATE` or `ZIP_CODE` or `LAST_LINE`.

This function needs to be called after you call `SetProperties`, but before you call `Update`.

See also

- [SetProperties](#)
- See the [CASService Properties](#) section for the enum names and purposes of the CASService properties.

## CASService Properties

The properties listed below are values within the CASServiceProperties structure. To set properties for your CASService process, instantiate a CASServiceProperties object, set the member properties listed below and pass this structure to the SetProperties() function.

We recommend that you set all property values. Any property that you do not set will contain the default value, which is usually false, 0 or the first enumeration value. However, this may not always be the case for all programming environments. So, to make sure that you receive the best results for your application, you should set every property.

**NOTE** For the following properties, we use the namespace MyCASServiceReference. This may be different on your system, depending on the development environment.

### CASS\_ABBREVIATE\_ADDRESS\_LINE

Data type

boolean

Description

Determines whether to abbreviate the address line to 30 characters.

Notes

- Address lines of less than 30 characters will not be abbreviated.
- This property affects the fields ADDRESS\_LINE\_1, ADDRESS\_LINE\_2 and ADDRESS\_BLOCK.

### CASS\_CERTIFY\_FLAG

Data type

CertifyFlagOption (an enumeration)

### Description

Determines if previously corrected addresses should be processed.

### Notes

- You must include the CASSDATE field as part of the SETTINGS\_FIELD\_LIST\_IN list.
- For large, regularly-processed lists, this property can improve processing speed significantly.
- Use one of the following values:
- CertifyFlagOption.eSkip – Check only those records not corrected with this issue.
- CertifyFlagOption.eCheckAllRecords – Check every record, regardless of date last corrected.
- CertifyFlagOption.eRebuild – Does not correct any records. Flags all records without ZIP + 4 Codes or corrected before the current issue as errors (error code set to 101). The CASSService then rebuilds the PS Form 3553 to only show as corrected those records that have been corrected with this issue.

## CASS\_DPV\_FAILURE\_AS\_ERROR

### Data type

boolean

### Description

Determines whether to treat addresses whose secondary address (apartment, suite, etc.) fails DPV processing as errors.

### Notes

Addresses that pass primary DPV processing but fail secondary checks can still be assigned a ZIP + 4 Code and qualify for presorted discounts.

## CASS\_DUAL\_ADDRESS\_INPUT\_PREFERENCE

### Data type

DualAddress (an enumeration)

### Description

Determines whether to use the street address or PO Box as the primary address for those addresses that have both.

### Notes

Use one of the following to set this property:

- DualAddress.eByPosition – The bottom address will be used, regardless of which type it is.
- DualAddress.ePreferPOBox – Uses PO Box addresses.
- DualAddress.ePreferStreet – Uses street addresses.

## CASS\_KEEP\_ALIAS\_ADDRESS

### Data type

boolean

### Description

Determines whether to keep the valid but unofficial street name alias given as input or to replace it with the official USPS street name.

### Notes

## CASS\_KEEP\_EXTRA\_PRIMARY\_DATA

### Data type

boolean

### Description

Determines whether to keep any additional and possibly extraneous information in the primary address line.

### Notes

## CASS\_LIST\_NAME

### Data type

string

**Description**

The name of the mailing list to process.

**Notes**

This value will be printed on the PS Form 3553.

**CASS\_LIST\_PROCESSOR****Data type**

string

**Description**

The name of the person processing this list with address correction.

**Notes**

This value will be printed on the PS Form 3553.

**CASS\_MAILERS\_ADDRESS****Data type**

string

**Description**

The street address of the person or business sending mail to this list.

**Notes**

This value will be printed on the PS Form 3553.

**CASS\_MAILERS\_CITY****Data type**

string

**Description**

The city of the person or business sending mail to this list.

**Notes**

This value will be printed on the PS Form 3553.

## CASS\_MAILERS\_NAME

### Data type

string

### Description

The name of the person or business sending mail to this list.

### Notes

This value will be printed on the PS Form 3553.

## CASS\_MAILERS\_STATE

### Data type

string

### Description

The state of the person or business sending mail to this list.

### Notes

This value will be printed on the PS Form 3553.

## CASS\_MAILERS\_ZIP

### Data type

string

### Description

The ZIP Code of the person or business sending mail to this list.

### Notes

This value will be printed on the PS Form 3553.

## CASS\_UPDATE\_UNCORRECTED\_CITY\_ST\_ZIP

### Data type

boolean

### Description

Determines if city, state and ZIP Code information should be updated for addresses that could not be fully corrected and validated.

## CASS\_USE\_SUITELINK

### Data type

boolean

### Description

This property is now obsolete. Suite<sup>Link</sup> is now a requirement of CASS processing.

### Notes

## FORMAT\_CASING

### Data type

Capitalization (an enumeration)

### Description

Determines the case in which addresses are returned.

### Notes

Use one of the following:

- Capitalization.eCapUppper – Upper case.
- Capitalization.eCapLower – Lower case.
- Capitalization.eCapMixed – Mixed case.

## FORMAT\_CITY

### Data type

AbbreviateCity (an enumeration)

### Description

Determines whether to abbreviate the city name.

## Notes

Use one of the following:

- `AbbreviateCity.eCityAbbrNever` – Always return the full city name.
- `AbbreviateCity.eCityAbbrForce` – Return the abbreviated city name if one exists.
- `AbbreviateCity.eCityAbbrInput` – THIS VALUE IS NO LONGER SUPPORTED.

## FORMAT\_FIRM\_OUTPUT

### Data type

`FirmOutput` (an enumeration)

### Description

Determines where to output a firm name in a corrected record.

## Notes

- This property applies to firm names found in corrected addresses, but not input as the `BUSINESS` field.
- Use one of the following:
- `FirmOutput.eFirmMoveToBusiness` – Return as the `BUSINESS` field only if that field was input as blank.
- `FirmOutput.eFirmMoveToBusinessOverwrite` – Always overwrite the `BUSINESS` field.

## FORMAT\_HIGHWAY\_CONTRACT

### Data type

`AddressElementFormat` (an enumeration)

### Description

Determines the format of returned highway contract addresses.

## Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_PMB\_OUTPUT

### Data type

PMBOutput (an enumeration)

### Description

Determines the location of the private mailbox number.

### Notes

Use one of the following:

- PMBOutput.ePMBWithUnits – Return on the same line as the unit information.

## FORMAT\_PO\_BOX

### Data type

AddressElementFormat (an enumeration)

### Description

Determines how to format a PO Box address.

### Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_POST\_DIRECTIONAL

### Data type

AddressElementFormat (an enumeration)

### Description

Determines how to format the returned post-directional, if any.

### Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_PRE\_DIRECTIONAL

### Data type

AddressElementFormat (an enumeration)

### Description

Determines how to format the returned pre-directional, if any.

### Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_PRIMARY\_ADDRESS\_OUTPUT

### Data type

PrimaryAddOutput (an enumeration)

### Description

Determines the location of the primary address information.

### Notes

Use one of the following:

- PrimaryAddOutput.eTopJustified – Return as ADDRESS\_LINE\_1 if ADDRESS\_LINE\_2 is blank. Else, return primary address data as ADDRESS\_LINE\_2 and secondary address data as ADDRESS\_LINE\_1.
- PrimaryAddOutput.eAddressLine2 – Always return as ADDRESS\_LINE\_2.

## FORMAT\_RURAL\_ROUTE

### Data type

AddressElementFormat (an enumeration)

### Description

Determines how to format returned rural route addresses.

### Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_SUFFIX

### Data type

AddressElementFormat (an enumeration)

### Description

Determines how to format the returned street suffix.

## Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_UNIT\_DESIGNATOR

### Data type

AddressElementFormat (an enumeration)

### Description

Determines how to format the returned unit type.

## Notes

Use one of the following:

- AddressElementFormat.eAbbreviation – Returns with standard USPS abbreviation.
- AddressElementFormat.eAbbrWithPunct – Returns with standard USPS abbreviation and punctuation.
- AddressElementFormat.eFullWord – Always returns full words.

## FORMAT\_UNIT\_OUTPUT

### Data type

UnitOutput (an enumeration)

### Description

Determines where to return the unit information for an address.

## Notes

Use one of the following:

- UnitOutput.ePrimaryAddressLine – Return at the end of the primary address line.
- UnitOutput.eSecondaryAddressLine – Return as the secondary address line if that line is blank.
- UnitOutput.eSecondaryAddressLineOverwrite – Overwrite the secondary address line with the unit information.

## FORMAT\_UPDATE\_CASE\_BUSINESS

### Data type

boolean

### Description

Determines if the casing specified by FORMAT\_CASING applies to the BUSINESS field.

## FORMAT\_UPDATE\_CASE\_NAMES

### Data type

boolean

### Description

Determines if the casing specified by FORMAT\_CASING applies to the FIRST\_NAME and LAST\_NAME fields.

### Notes

## SETTINGS\_FIELD\_LIST\_IN

### Data type

ArrayOfField

### Description

An array of fields that will be included in every input address record.

### Notes

- At minimum, this array should include ADDRESS\_LINE\_1 plus either CITY and STATE, ZIP\_CODE or LAST\_LINE.

- See the Fields section of this document for a list of fields that can be included.

## SETTINGS\_FIELD\_LIST\_OUT

### Data type

ArrayOfField

### Description

An array of fields that will compose each returned address record.

### Notes

See the Fields section of this document for a list of fields that can be included.

## SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT

### Data type

int

### Description

Indicates the number of records that will be sent and returned with each Update call.

### Notes

You must set this property.

## SETTINGS\_RECORD\_COUNT

### Data type

int

### Description

Indicates the total number of records to be processed across all Update calls.

### Notes

You must set this property.

## CASService Fields

The items below are part of the Field enumeration. Include these names in the ArrayOfFields passed to either SETTINGS\_FIELD\_LIST\_IN or SETTINGS\_FIELD\_LIST\_OUT to determine which fields

CASService looks to process from incoming addresses or returns in processed records. All field values are strings.

Many of these fields are output only. The descriptions below indicate which fields do not accept input.

For an effective CASService process, we recommend that you use at least the following fields as input: ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE, and ZIP\_CODE.

**NOTE** For the following, we use the namespace CASServiceReference. This may be different on your system, depending on the development environment.

The following fields must be in the list of input fields if you want to include them in the list of output fields:

- RECORD\_ID
- All USER\_DEFINED fields
- COUNTRY
- NAME\_SALUTATION
- MIDDLE\_NAME
- NAME\_SUFFIX

## ADDRESS\_BLOCK

### Description

The full address as would be printed on the front of a mail piece.

### Notes

You can pass or retrieve this value instead of ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE, and ZIP\_CODE.

## ADDRESS\_LINE\_1

### Description

The first line of this address.

### Notes

We recommend that you set this field for each address.

This will be the primary address information unless you have unit information on a separate line.

## ADDRESS\_LINE\_2

### Description

The second line of this address.

### Notes

- We recommend that you set this field for each address.
- If you have unit information on a separate line, this may contain the street address.

## BUSINESS

### Description

The name of the business, if any.

### Notes

Depending on the value of `FORMAT_FIRM_OUTPUT`, Suite<sup>Link</sup> may return this information for a corrected address.

## CARRIER\_ROUTE

### Description

The carrier route for this address.

### Notes

- This will be returned for a successfully corrected address.
- Output only.

## CASSDATE

### Description

An encoded string that contains the date that this record was last processed.

### Notes

Output only.

## CITY

### Description

The city of this address.

### Notes

We recommend that you set this field for each address.

## CITY\_ABBREVIATED

### Description

The city name, abbreviated to 30 characters.

### Notes

Output only.

## CONGRESSIONAL\_DISTRICT

### Description

The congressional district for this address.

### Notes

Output only.

## COUNTRY

### Description

The country for this address.

### Notes

For most addresses, this will be United States.

## COUNTY\_CODE

### Description

The five-digit county code for this address.

### Notes

Output only.

## COUNTY\_NAME

### Description

The name of the county in which this address resides.

## DPC

### Description

The delivery point/check digit for this address.

### Notes

Output only.

## DPV\_CODED

### Description

Indicated if DPV processing confirmed the deliverability of this address.

### Notes

Output only.

## DPV\_FOOTNOTE

### Description

One or more two character strings that indicate the results of DPV processing.

### Notes

Output only.

Returns one of the following:

- LK – Processing locked out due to a seed record being processed.
- AA – Matched to the ZIP+4 file
- A1 – No match against the ZIP+4 file
- BB – Matched to DPV file (all components confirmed)
- CC – Matched only after removing secondary information; they were presented but invalid.

- N1 – Input primary matched, but high-rise missing secondary number.
- M1 – Primary number missing.
- M3 – Primary number invalid.
- P1 – Input missing PO, RR or HC box number.
- P3 – Failed DPV because of invalid PO, RR or HC box number.
- RR – Matched CMRA (found in CMRA file).
- R1 – Matched CMRA, but secondary number (i.e., PMB) missing.
- U1 – Matched unique ZIP Code.
- G1 – Matched general delivery.
- F1 – Matched military address.

## DPV\_INDICATOR

### Description

Indicates how detailed the DPV match is.

### Notes

Output only.

Returns one of the following:

- Y – Both the primary and secondary (if present) validated against the DPV database.
- S – The primary address is valid according to DPV, but the secondary is invalid.
- D – The primary address is valid according to DPV, but the address is missing secondary information.
- N – The primary address is not valid according to DPV.
- "" – The address was not presented to the DPV table, because it was missing components needed for the lookup. This usually means the record is not ZIP+4 coded.

- X – The DPV database has been locked-out because of a protocol violation; you must unlock DPV before any more addresses will be presented to the DPV table.
- E – The DPV data file is more than 105 days old; by USPS restrictions, no more addresses can be presented to the DPV table.

## DPV\_IS\_CMRA

### Description

Indicates if the address is a commercial mail-receiving agent.

### Notes

Output only.

## DPV\_IS\_NOSTAT

### Description

Indicates whether the address is a new construction that does not yet have regular mail delivery.

### Notes

Output only.

## DPV\_IS\_VACANT

### Description

Indicates if this address has been unoccupied for at least 90 days.

### Notes

Output only.

## DP\_BARCODE

### Description

The delivery point barcode for this address.

### Notes

Output only.

Apply the SATBAR font, size 12, to this string to print a POSTNET barcode.

**NOTE** As of January 27, 2013, POSTNET barcodes do not qualify mailings for Automation

| discounts.

## ERROR\_CODE

### Description

An error code for this address.

### Notes

Every address will receive an error code, regardless of whether it was corrected successfully or not.

Output only.

See the Error Code table in the Appendix for a listing of all error codes and descriptions.

## ERROR\_STRING

### Description

A description of the error code for this address.

### Notes

Every address will receive an error code, regardless of whether it was corrected successfully or not.

Output only.

See the Error Code table in the Appendix for a listing of all error codes and descriptions.

## EWS\_CODED

### Description

Indicates that an address is on a new street not yet included in the USPS databases.

### Notes

Output only.

## EXTRA\_INFO

### Description

Contains any extra information from the input address that the CASSService was not able to parse into an address field.

#### Notes

Output only.

### FIRST\_NAME

#### Description

The first name of the resident of this address.

#### Notes

This field will not be processed and will be returned the same as input.

### GEOCODE\_CENSUS\_BLOCK

#### Description

The census block for this address.

#### Notes

A census block is the smallest area for which the census bureau produces information.

Output only.

You must have the Geocode add-on installed to generate this information.

### GEOCODE\_CENSUS\_TRACT

#### Description

The census tract in which an address is located.

#### Notes

A census tract is a geographic area, often coinciding with the limits of cities or towns, defined for census purposes. A census tract can contain several census blocks.

Output only.

You must have the Geocode add-on installed to generate this information.

### GEOCODE\_FOOTNOTE

#### Description

Indicates the results of Geocode processing.

#### Notes

Requires the Geocode add-on.

Output only.

Contains one of the following:

- 00 – Unable to perform a geocode lookup.
- 03 – Geocode data based on a 3-digit ZIP Code.
- 05 – Geocode data based on a 5-digit ZIP Code.
- 07 – Geocode data based on a 7-digit ZIP Code.
- 09 – Geocode data based on a 9-digit ZIP Code.

## GEOCODE\_LATITUDE

### Description

The latitude of this address.

### Notes

Output only.

You must have the Geocode add-on installed to generate this information.

## GEOCODE\_LONGITUDE

### Description

The longitude of this address.

### Notes

Output only.

You must have the Geocode add-on installed to generate this information.

## GEOCODE\_MSA\_CODE

### Description

Contains the Metropolitan Statistical Area code for this address.

### Notes

Output only.

You must have the Geocode add-on installed to generate this information.

## IS\_RESIDENCE

### Description

Indicates whether this address is a residence or not.

### Notes

To use this property, you must have the RDI add-on and subscribe to the RDI service from the USPS.

Output only.

## LACS\_CODED

### Description

Indicates whether this address was flagged by LACS<sup>Link</sup> processing.

### Notes

Output only.

## LACS\_FOOTNOTE

### Description

Provides additional information about the LACS process.

### Notes

Output only.

Contains one of the following:

- blank – Not processed / Seed record.
- 00 – No match.
- 09 – Matched to default high-rise address; address not updated.
- 14 – Match failed to build new address.
- 92 – Match secondary dropped from input.
- A – Match success.

## LACS\_INDICATOR

### Description

Indicates the results of a LACS process.

### Notes

Output only.

Returns one of the following:

- blank – Not processed.
- N – Match, but there was a failure to build new address.
- Y – Match success, new address provided.
- S – Match with secondary dropped from input.
- F – Seed record.

## LAST\_LINE

### Description

The last line of the address block.

### Notes

You can set or retrieve this field instead of CITY, STATE and ZIP\_CODE.

## LAST\_NAME

### Description

The last name of the resident of this address.

### Notes

This field will not be processed and will be returned the same as input.

## LOT\_NUMBER

### Description

The line-of-travel number for an address.

#### Notes

This field is used in carrier route sorts.

Output only.

### MATCHED\_TO\_DEFAULT

#### Description

Indicates whether this address was matched to a default ZIP Code.

#### Notes

Default ZIP Codes are only five digits in length.

Output only.

### PMB\_NUMBER

#### Description

The private mailbox number for this address.

#### Notes

### POST\_DIRECTIONAL

#### Description

The post-directional indicator for this address.

#### Notes

### PRE\_DIRECTIONAL

#### Description

The pre-directional indicator for this address.

#### Notes

### RECORD\_TYPE

#### Description

Indicates the type of address that this is.

#### Notes

Contains one of the following characters:

- S – Street record
- P – Post office box
- R – Rural Route or Highway Contract
- H – High-rise, Building or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

## SKIPPED\_CERTIFY

### Description

Indicates whether this address was skipped during CASS processing.

### Notes

Output only.

## STATE

### Description

The state for this address.

## STREET\_NAME

### Description

The name of the street of this address.

## SUFFIX

### Description

The street suffix of this address.

### Notes

For example, St., Ave., or Ln.

## SUITELINK\_FOOTNOTE

### Description

Indicates the results of Suite<sup>Link</sup> processing.

### Notes

Output only.

Returns one of the following:

- "" – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup within the Suite<sup>Link</sup> file. Only default high-rise addresses qualify for a Suite<sup>Link</sup> lookup.
- A – The address was processed and secondary information was added to the resulting address.
- 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.

## UNIT\_DESIGNATOR

### Description

The type of unit for the secondary address data.

### Notes

For example, Ste., Apt., or Fl.

## UNIT\_NUMBER

### Description

The unit number for the secondary address data.

### Notes

## URBANIZATION

### Description

The urbanization of this address.

### Notes

Puerto Rico only.

## ZIPCODE

### Description

The ZIP Code of this address.

## The Web Services MOVEService Interface for Updating Moved Addresses

MOVEService is a Web service interface to the MOVE Task library. BCC Software provides this interface as an alternative to the COM and .NET interfaces. This service takes a list of addresses, corrects them using a CASS Certified address correction process, matches them against the USPS NCOA<sup>Link</sup> database, and returns them to you.

The MOVEService is flexible enough where you can control the amount of information you receive and the format in which it arrives. Afterwards, you can save both the Address Correction Report (PS Form 3553) and the NCOA<sup>Link</sup> Processing Report as a single PDF file.

To create an application that uses the MOVEService to update your records with change-of-address information, you must have a valid BCC Architect registration key and a registered Move Update account. You will also need Move Update credits for your job. Each address processed uses one credit. Contact your account representative for more information.

MOVEService requires that your implementation be able to store and use cookies. Because the MOVEService processing occurs on a remote server and requires that you make multiple function calls to this server, you need some way of maintaining a connection to the job you are running. Cookies will allow you to maintain this continuity with the server.

When using a .NET client, create an instance of the System.Net.CookieContainer class and attach it to your MoveService object. This creates a place to store the session ID cookie that is returned. The following C# sample code shows this:

### EXAMPLE

```
MoveService mService = new MoveService();
if (mService.CookieContainer == null)
    mService.CookieContainer = new System.Net.CookieContainer();
```

The following sample code shows you how to do this in Java:

### EXAMPLE

```
URL wsdlURL = MoveService.WSDL_LOCATION;
MoveService ms = new MoveService(wsdlURL, MoveService.SERVICE);
```

```
MoveServiceSoap move = ms.getMoveUpdateServiceSoap12();  
// setup to preserve the session cookie  
( (BindingProvider)move).getRequestContext().put  
(BindingProvider._SESSION_MAINTAIN_PROPERTY, Boolean.TRUE);
```

## MOVEService Overview

The following general procedure should be used to implement the MOVEService:

1. Add a Web reference to the MOVEService. The address of the service is <https://ws.sat-orisoftware.com/Architect/US/Move/MoveService.asmx>.
2. Create a MOVEService object.
3. Call PrepareTask() on your registration key. This will return a session ID.
4. Create one or more MoveService.AddressBlock object that contain your address records. An AddressBlock object contains an array of MoveService.Address objects. For each Address object, add at least the fields FIRST\_NAME, LAST\_NAME, BUSINESS, ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE and ZIP\_CODE. Add any other fields that you want to pass to the MOVEService.
5. Create arrays of MoveService.Field objects that define the sets of input and output data fields. The input field list should match those you added to the block of addresses in the previous step. The output field list can be any or all of the input fields, plus any of the output data fields that the MOVEService produces. See the Fields section of this document for the full list.
6. Call SetProperties to set the desired properties. Add the individual properties to set to a MoveService.MoveServiceProperties enumeration. You must set SETTINGS\_FIELD\_LIST\_IN, SETTINGS\_FIELD\_LIST\_OUT, LOGIN\_CUSTOMER\_ID, LOGIN\_CUSTOMER\_PASSWORD and SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT.
7. Call ValidateProperties.
8. Call Send on the AddressBlock object or objects you created in step 4.
9. Call DoProcess().
10. Call GetProcessStatus() to check the state of the process. While the state.phase is Processing, wait.

11. When the state.phase is Complete, call Retrieve. This will return an updated AddressBlock object.
12. Iterate through the AddressBlock object to get each updated Address object. In turn, iterate through each Address object to get the updated fields for each record. The fields for each output record will match those specified by the SETTINGS\_FIELD\_LIST\_OUT property set earlier.
13. Repeat steps 11 and 12 until you have retrieved all records.
14. Call GetReportsAsPDF() to return the CASS Summary Report (PS Form 3553) and NCOA<sup>Link</sup> Processing Summary Report as a byte array, which you can save on your local system.
15. Call EndTask and clean up any other resources used.

## MOVEService Functions

Below are the methods, properties and fields available in MOVEService. If you have added a Web reference to the MOVEService in Visual Studio, you can view all of these functions in the Object Browser.

**NOTE** For the following functions, we use the namespace MoveServiceReference. This may be different on your system, depending on the development environment.

### DoProcess

#### Syntax

```
void DoProcess();
```

#### Description

Starts processing all the addresses that you have sent to the MOVEService.

#### Parameters

None.

#### Return values

None.

#### Notes

This function does not return anything, as processing may take a significant amount of time.

Call GetProcessStatus() to return the current state of the process.

See also

[GetProcessStatus\(\)](#)

## EndTask

Syntax

```
void EndTask();
```

Description

Formally ends the MOVEService process and cleans up any resources that it used.

Parameters

None.

Return values

None.

Notes

Call this when you are finished processing addresses and you have called Retrieve() for all of your blocks of addresses.

See also

[Retrieve\(\)](#)

## GetProcessStatus

Syntax

```
ProcessingStatus GetProcessStatus();
```

Description

Returns the current state of the MOVEService processing.

Parameters

None.

Return values

A ProcessingStatus object. This object contains two values, phase and recordsProcessed. The recordsProcessed variable is an integer that indicates how many records have been processed.

The phase object contains a `MoveServiceReference.ProcessingState` object, which will be one of the following:

- Processing – The process is still in progress, so please wait and get the status again.
- Complete – The process is now finished and you can call `Retrieve()` to get the updated addresses.
- Error – The MOVEService was unable to complete the process.

#### Notes

While this function returns Processing as the phase value, do not call `Retrieve()`. Doing so will return an error.

#### See also

- [Retrieve\(\)](#)

## GetProperties

#### Syntax

```
MoveServiceReference.MoveServiceProperties GetProperties(MoveServiceReference.ArrayofProperty PropertyIDs);
```

#### Description

Returns the values of the Property items passed to it.

#### Parameters

A Property array that contains the properties for which you want to get the values.

#### Return values

A `MoveServiceProperties` object that contains the properties specified by the Property array passed to it.

#### Notes

The returned object will only contain those properties specified in the parameter array.

Any properties that you have not already set will return the default values.

See also

[ValidateProperties\(\)](#)

[SetProperties\(\)](#)

The [MoveServiceProperties](#) list.

## GetReportPDF

Syntax

```
byte[] GetReportPDF();
```

Description

Returns the Address Correction Report and NCOA<sup>Link</sup> Processing Report as a byte array that can be saved on the local machine as a PDF file.

Parameters

None.

Return values

A byte array that can be saved on the local machine as a PDF file.

Notes

To open a PDF file, use Adobe Reader, freely available from the Web site of Adobe Software.

## PrepareTask

Syntax

```
string PrepareTask(string RegistrationKey);
```

Description

Establishes a session with the MOVEService using your registration key.

Parameters

Your BCC Architect registration key as a string.

Return values

A session ID as a string.

## Notes

This method must be called before you can set any properties or process any addresses.

Call this function only once, after you create the MOVEService object.

You may need your session ID if you are disconnected from the service. Call ReconnectTask() to reestablish a connection with the MOVEService.

## See also

ReconnectTask()

## Retrieve

### Syntax

```
MoveServiceReference.AddressBlock Retrieve ();
```

### Description

Returns the updated addresses once processing is complete.

### Parameters

None.

### Return values

An AddressBlock object that contains processed addresses. The AddressBlock object contains an array of Address objects.

## Notes

The addresses in the AddressBlock object will have all the fields specified in the SETTINGS\_FIELD\_LIST\_OUT property.

Each address block will contain a number of addresses specified by the property, SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT.

You may need to call this function several times to return all of your processed addresses.

## See also

- [SETTINGS\\_FIELD\\_LIST\\_OUT](#)
- [SETTINGS\\_INPUT\\_BLOCK\\_RECORD\\_COUNT](#)

- [Send\(\)](#)

## Send

### Syntax

```
void Send(AddressBlock AddressBlock);
```

### Description

Sends the specified block of addresses to the MOVEService to be processed.

### Parameters

An AddressBlock object that contains the addresses you wish to process.

### Return values

None.

### Notes

You can send up to 500 addresses in each address block. You may want to send fewer to improve processing speed.

Call this function for every block of addresses you wish to process.

Call DoProcess() to start processing these addresses.

### See also

- [DoProcess\(\)](#)
- [Retrieve\(\)](#)

## SetProperties

### Syntax

```
void SetProperties(MoveServiceReference.MoveServiceProperties  
MoveProps);
```

### Description

Sets the properties for the MOVEService process using a MoveServiceProperties enumeration.

### Parameters

A MoveServiceProperties object that contains the settings you would like to apply to this process.

### Return values

None.

### Notes

You can set as many properties at once as you wish.

You must set SETTINGS\_FIELD\_LIST\_IN, SETTINGS\_FIELD\_LIST\_OUT, LOGIN\_CUSTOMER\_ID, LOGIN\_CUSTOMER\_PASSWORD and SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT.

Once you have set the properties, you must then call ValidateProperties().

### See also

- [ValidateProperties\(\)](#)
- [GetProperties\(\)](#)
- The [MoveServiceProperties](#) list.

## ValidateProperties

### Syntax

```
Bool ValidateProperties();
```

### Description

Validates the properties that you have passed to the service.

### Parameters

None.

### Return values

Returns true if the properties sent are valid.

### Notes

You must call this function before DoProcess().

This function ensures that you have set all of the necessary properties.

See also

- [SetProperties\(\)](#)
- [GetProperties\(\)](#)
- The [MoveServiceProperties](#) list.

## MOVEService Properties

Use the following properties to either configure the MOVEService using `SetProperties(MoveServiceProperties)` or to retrieve the existing settings using `GetProperties(MoveServiceProperties)`. The `MoveServiceProperties` is an enumeration that can contain any or all of these properties.

Below the name of the property is the data type it takes or returns. All properties can be set or retrieved. Any properties that you attempt to retrieve before you assign a value to them will return the default value. When processing addresses with the MOVEService, you must set `SETTINGS_FIELD_LIST_IN`, `SETTINGS_FIELD_LIST_OUT`, `LOGIN_CUSTOMER_ID`, `LOGIN_CUSTOMER_PASSWORD` and `SETTINGS_INPUT_BLOCK_RECORD_COUNT`.

**NOTE** For the following properties, we use the namespace `MoveServiceReference`. This may be different on your system, depending on the development environment.

### CASS\_DPV\_FAILURE\_AS\_ERROR

Data Type

string

Description

Determines whether the service treats DPV failures that would otherwise allow an address to receive a ZIP + 4 Code as errors.

Notes

While DPV is required for CASS processing, addresses that fail DPV because of missing or invalid secondary information will still pass CASS processing.

Set this property to 1 to enable it.

### CASS\_DUAL\_ADDRESS\_INPUT\_PREFERENCE

Data Type

string

### Description

Determines whether to give the street address or PO Box preference for addresses that have both.

### Notes

Use the following to set this property:

- 0 – The bottom address will be used, regardless of which type it is.
- 1 – Uses PO Box addresses.
- 2 – Uses street addresses.

## CASS\_KEEP\_ALIAS\_ADDRESS

### Data Type

string

### Description

Determines whether an input address with a valid street name alias will be allowed or whether they will be replaced with the official USPS street name.

### Notes

Set this property to 1 to enable it.

## CASS\_KEEP\_EXTRA\_PRIMARY\_DATA

### Data Type

string

### Description

Determines whether to keep any additional and possibly extraneous information in the primary address line.

### Notes

Set this property to 1 to enable it.

## CASS\_LIST\_NAME

### Data Type

string

**Description**

The name of the list being processed.

**Notes**

This name will print on the Address Correction report.

**CASS\_LIST\_PROCESSOR****Data Type**

string

**Description**

The name of the person processing the list.

**Notes**

This will be printed on the Address Correction Report.

**CASS\_MAILERS\_ADDRESS****Data Type**

string

**Description**

The address of the company or individual doing the mailing.

**Notes**

This will be printed on the Address Correction Report.

**CASS\_MAILERS\_CITY****Data Type**

string

**Description**

The city of the company or individual doing the mailing.

**Notes**

This will be printed on the Address Correction Report.

## CASS\_MAILERS\_NAME

### Data Type

string

### Description

The name of the company or individual doing the mailing.

### Notes

This will be printed on the Address Correction Report.

## CASS\_MAILERS\_STATE

### Data Type

string

### Description

The state in which the company or individual doing the mailing resides.

### Notes

This will be printed on the Address Correction Report.

## CASS\_MAILERS\_ZIP

### Data Type

string

### Description

The ZIP Code of the company or individual doing the mailing.

### Notes

This will be printed on the Address Correction Report.

## CASS\_UPDATE\_UNCORRECTED\_CITY\_ST\_ZIP

### Data Type

boolean

### Description

Determines whether the corrected city, state and/or ZIP Code should be returned for addresses that otherwise could not be corrected.

### Notes

## FORMAT\_CASING

### Data Type

Capitalization

### Description

Sets the capitalization format for processed addresses.

### Notes

Use one of the following:

- MoveServiceReference.Capitalization.eCapUpper – Sets to all UPPER CASE.
- MoveServiceReference.Capitalization.eCapLower – Sets to all lower case.
- MoveServiceReference.Capitalization.eCapMixed – Sets to Mixed Case.

## FORMAT\_CITY

### Data Type

AbbreviateCity

### Description

Determines whether to abbreviate the city returned from the MOVEService process.

### Notes

Use one of the following:

- MoveServiceReference.AbbreviateCity.eCityAbbrNever – The city will not be abbreviated.
- MoveServiceReference.AbbreviateCity.eCityAbbrForce – The city will always be abbreviated, if possible.

- MoveServiceReference.AbbreviateCity.eCityAbbrInput – The output city will be abbreviated only if the input city was.

## FORMAT\_FIRM\_OUTPUT

### Data Type

FirmOutput

### Description

Determines how to return a firm name found with an address record.

### Notes

Use one of the following:

- MoveServiceReference.FirmOutput.eFirmMoveToBusiness – If the input BUSINESS field was blank, return field information in the BUSINESS field.
- MoveServiceReference.FirmOutput.eFirmMoveToBusinessOverwrite – Always output the firm information as the BUSINESS field.

## FORMAT\_HIGHWAY\_CONTRACT

### Data Type

AddressElementFormat

### Description

Determines how the service formats Highway Contract address information.

### Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

## FORMAT\_PMB\_OUTPUT

### Data Type

PMBOutput

### Description

Determines the output location of personal mailbox information, if any.

### Notes

Use one of the following:

- MoveServiceReference.PMBOutput.ePMBWithUnits – PMB information outputs on the same line as the unit information.

## FORMAT\_PO\_BOX

### Data Type

AddressElementFormat

### Description

Determines how the service will output PO Box information.

### Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

## FORMAT\_POST\_DIRECTIONAL

### Data Type

AddressElementFormat

### Description

Determines how the service will format post-directional information.

### Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

## FORMAT\_PRE\_DIRECTIONAL

### Data Type

AddressElementFormat

### Description

Determines how the service will format pre-directional information.

### Notes

Set this to one of the following:

- MoveServiceReference.AddressElementFormat.eAbbreviation – Abbreviates the information.
- MoveServiceReference.AddressElementFormat.eAbbrWithPunct – Abbreviates the information using punctuation.
- MoveServiceReference.AddressElementFormat.eFullWord – Outputs the information without abbreviations.

## FORMAT\_PRIMARY\_ADDRESS\_OUTPUT

### Data Type

PrimaryAddOutput

### Description

Determines how the service will output the primary address information.

### Notes

Set this to one of the following:

- `MoveServiceReference.PrimaryAddOutput.eTopJustified` – Returns primary address data as `ADDRESS_LINE_1` unless the input address contains secondary information in `ADDRESS_LINE_2`. In this case, the secondary information will be returned as `ADDRESS_LINE_1` and the primary address data will be returned in `ADDRESS_LINE_2`.
- `MoveServiceReference.PrimaryAddOutput.eAddressLine2` – Always returns the primary address data as `ADDRESS_LINE_2`.

## FORMAT\_RURAL\_ROUTE

### Data Type

`AddressElementFormat`

### Description

Determines how the service will format rural route address information.

### Notes

Set this to one of the following:

- `MoveServiceReference.AddressElementFormat.eAbbreviation` – Abbreviates the information.
- `MoveServiceReference.AddressElementFormat.eAbbrWithPunct` – Abbreviates the information using punctuation.
- `MoveServiceReference.AddressElementFormat.eFullWord` – Outputs the information without abbreviations.

## FORMAT\_SUFFIX

### Data Type

`AddressElementFormat`

### Description

Determines how the service will format street suffix information.

### Notes

Set this to one of the following:

- `MoveServiceReference.AddressElementFormat.eAbbreviation` – Abbreviates the information.
- `MoveServiceReference.AddressElementFormat.eAbbrWithPunct` – Abbreviates the information using punctuation.
- `MoveServiceReference.AddressElementFormat.eFullWord` – Outputs the information without abbreviations.

## FORMAT\_UNIT\_DESIGNATOR

### Data Type

`AddressElementFormat`

### Description

Determines how the service will format unit type information, such as Suite or Apartment.

### Notes

Set this to one of the following:

- `MoveServiceReference.AddressElementFormat.eAbbreviation` – Abbreviates the information.
- `MoveServiceReference.AddressElementFormat.eAbbrWithPunct` – Abbreviates the information using punctuation.
- `MoveServiceReference.AddressElementFormat.eFullWord` – Outputs the information without abbreviations.

## FORMAT\_UNIT\_OUTPUT

### Data Type

`UnitOutput`

### Description

Determines where the service outputs the secondary address information, such as the unit or apartment number.

### Notes

In the event this conflicts with `FORMAT_PRIMARY_ADDRESS_OUTPUT`, the primary format will take precedence.

Set this to one of the following:

- `MoveServiceReference.UnitOutput.ePrimaryAddressLine` – Appends this data to the primary address information. This is the default for this property.
- `MoveServiceReference.UnitOutput.eSecondaryAddressLine` – Returns this data as `ADDRESS_LINE_2` only if that field was blank.
- `MoveServiceReference.UnitOutput.eSecondaryAddressLineOverwrite` – Always returns this data as `ADDRESS_LINE_2` and overwrites any existing data there.

## FORMAT\_UPDATE\_CASE\_BUSINESS

### Data Type

boolean

### Description

Determines whether to apply the casing specified by `FORMAT_CASING` to the `BUSINESS` field.

### Notes

## FORMAT\_UPDATE\_CASE\_NAMES

### Data Type

boolean

### Description

Determines whether to apply the casing specified by `FORMAT_CASING` to the `FIRST_NAME` and `LAST_NAME` fields.

#### Notes

### LOGIN\_ADMIN\_ID

#### Data Type

string

#### Description

The login ID if you have an administrator account.

#### Notes

In order to run the MOVEService if you have a list administrator account, you must set this property to the login ID that BCC Software has supplied to you.

### LOGIN\_ADMIN\_PASSWORD

#### Data Type

string

#### Description

The password if you have an administrator account.

#### Notes

In order to run the MOVEService if you have a list administrator account, you must set this property to the password that BCC Software has supplied to you.

### LOGIN\_BROKER\_ID

#### Data Type

string

#### Description

The login ID if you have a list broker account.

#### Notes

In order to run the MOVEService if you have a list broker account, you must set this property to the login ID that BCC Software has supplied to you.

## LOGIN\_BROKER\_PASSWORD

### Data Type

string

### Description

The password if you have a list broker account.

### Notes

In order to run the MOVEService if you have a list broker account, you must set this property to the password that BCC Software has supplied to you.

## LOGIN\_CUSTOMER\_ID

### Data Type

string

### Description

The login ID if you have a customer account.

### Notes

In order to run the MOVEService if you have a client account, you must set this property to the login ID that BCC Software has supplied to you.

## LOGIN\_CUSTOMER\_PASSWORD

### Data Type

string

### Description

The password if you have a customer account.

### Notes

In order to run the MOVEService if you have a client account, you must set this property to the password that BCC Software has supplied to you.

## MOVE\_BUYER\_NAME

### Data Type

string

#### Description

The name of the client who owns the list being processed.

#### Notes

This property only applies to administrator or list broker accounts.

### MOVE\_CLIENT\_ID\_LIST

#### Data Type

String

#### Description

Returns a list of client IDs and their PAF expiration dates as a single string.

#### Notes

The format is [CLIENT\_ID],YYYY-MM-DD. Each ID-date pair is separated from others by a carriage return.

Before you can retrieve this data, you must set LOGIN\_BROKER\_ID or LOGIN\_CUSTOMER\_ID first, then call ValidateProperties.

The default value is an empty string.

### MOVE\_CUSTOMER\_MAILERID

#### Data Type

string

#### Description

Your mailer ID that the USPS assigned to you.

#### Notes

This will be printed on the NCOA [Link](#) Processing Form.

### MOVE\_HIGH\_MATCH\_RATE\_REASON

#### Data Type

string

## Description

If you have a mailing list that produces an unusually high rate of matches to changed address entries, then you should provide a reason why this occurs.

## Notes

You will only need to set this property if your list was previously processed by ANK or another similar tool.

This text will be printed on the NCOA<sup>Link</sup> Processing Report if necessary.

## MOVE\_MAIL\_CLASS

### Data Type

string

### Description

The mail class by which you intend to send the processed mailing.

### Notes

Use the following:

- A – First Class Only
- B – Periodicals Only
- C – Standard Mail Only
- D – Package Services Only
- E – First Class And Periodicals
- F – First Class And Standard
- G – First Class And Package
- H – Periodicals And Standard
- I – Periodicals And Package
- J – Standard And Package

- K – First Class, Periodicals, Standard
- L – First Class, Periodicals, Package
- M – First Class, Standard, Package
- N – Periodicals, Standard, Package
- O – All (default)

## MOVE\_MATCH\_FLAG

### Data Type

string

### Description

Determines what type of moves the service will search for.

### Notes

Use one of the following:

- S = Standard. Business, Individual and Family. This is the default value.
- C= Business and Individual
- B= Business only
- I= Individual only
- R=Individual and Family

## MOVE\_MOVE\_MONTH\_RANGE

### Data Type

string

### Description

Determines how the maximum age of the move for which the service will search.

## Notes

You may search for moves that are up to 48 months old.

Set this property to the maximum number of months in which to search.

## MOVE\_MULTI\_NAME\_HANDLE

### Data Type

string

### Description

Determines how the service treats records with multiple names.

## Notes

Use one of the following:

- 0 – Search only if a common last name is found.
- 1 – Search using all names found.
- 2 – Skip the address.

## SETTINGS\_FIELD\_LIST\_IN

### Data Type

ArrayOfField

### Description

An array of MoveServiceReference.Field items that identify which fields the input addresses will contain.

## Notes

This is required to run the MOVEService.

You must add at least ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE and ZIP\_CODE.

Depending on the type of moves you are searching for, you will also need one or more of the following: FIRST\_NAME, LAST\_NAME and BUSINESS.

## SETTINGS\_FIELD\_LIST\_OUT

### Data Type

ArrayOfField

### Description

An array of MoveServiceReference.Field items that determine which fields the processed output addresses will contain.

### Notes

This is required to run the MOVEService.

We recommend that you add at least ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE and ZIP\_CODE.

## SETTINGS\_INPUT\_BLOCK\_RECORD\_COUNT

### Data Type

int

### Description

Set this equal to the number of addresses you send per block of addresses.

### Notes

Each address block should contain this number of records, except for the last block, which can contain less.

You must set this property in order to run a MOVEService job.

## SETTINGS\_RECORD\_COUNT

### Data Type

int

### Description

Set this equal to the total number of records in your mailing list.

### Notes

This property is required if you send multiple blocks of addresses.

## MOVEService Fields

The items below are part of the Field enumeration. Include these names in the ArrayOfFields passed to either SETTINGS\_FIELD\_LIST\_IN or SETTINGS\_FIELD\_LIST\_OUT to determine which fields MOVEService looks to process from incoming addresses or returns in processed records. All field values are strings.

Many of these fields are output only. The descriptions below will indicate which fields do not accept input. All fields with the prefixes BEFORE\_CASS, AFTER\_CASS or AFTER\_NCOA do not accept input.

For an effective MOVEService process, we recommend that you use at least the following fields as input: ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, STATE and ZIP\_CODE. Depending on the type of moves you are searching for, you will also need one or more of the following: FIRST\_NAME, LAST\_NAME and BUSINESS.

**NOTE** For the following fields, we use the namespace MoveServiceReference. This may be different on your system, depending on the development environment.

The following fields must be in the list of input fields if you wish to include them in the list of output fields:

- RECORD\_ID
- All USER\_DEFINED fields
- COUNTRY
- NAME\_SALUTATION
- MIDDLE\_NAME
- NAME\_SUFFIX

### RECORD\_ID

#### Description

A number that identifies a record.

## Notes

- This will not be generated during a MOVEService process.
- You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.

## FIRST\_NAME

### Description

The first name of the individual associated with this record.

### Notes

This will be used for Individual matches.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_FIRST\_NAME
- AFTER\_NCOA\_MATCH\_FIRST\_NAME – This is the first name of the matched record in the NCOA<sup>Link</sup> database.

## LAST\_NAME

### Description

The last name of the individual associated with this record.

### Notes

This will be used for Individual and Family matches.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_LAST\_NAME
- AFTER\_NCOA\_MATCH\_LAST\_NAME – This is the last name of the matched record in the NCOA<sup>Link</sup> database.

## NAME\_SALUTATION

### Description

A greeting (Mr., Mrs, Ms.) for the name on this record.

## Notes

You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.

Has the following related field that shows this data before CASS processing:

- BEFORE\_CASS\_PREFIX\_TITLE

## MIDDLE\_NAME

### Description

The middle name of the person on this record.

## Notes

You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_MIDDLE\_NAME
- AFTER\_NCOA\_MIDDLE\_INITIAL

## NAME\_SUFFIX

### Description

The suffix (Jr., Sr., etc.), if any, for the person on this record.

## Notes

You must pass this as input to retrieve it as output. MOVEService will throw an exception otherwise.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_SUFFIX\_TITLE
- AFTER\_NCOA\_MATCH\_SUFFIX\_NAME

## BEFORE\_CASS\_FULL\_NAME

### Description

The full name of this record before CASS processing.

### Notes

This is assembled from the FIRST\_NAME and LAST\_NAME fields.

## BUSINESS

### Description

The name of the business associated with this address.

### Notes

This will be used for Business matches.

Has the following related field that shows this data before CASS processing:

- BEFORE\_CASS\_BUSINESS

## ADDRESS\_LINE\_1

### Description

The first line of this address.

### Notes

This will be the primary address information unless you have unit information on a separate line.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_ADDRESS\_LINE\_1
- AFTER\_CASS\_ADDRESS\_LINE\_1
- AFTER\_NCOA\_ADDRESS\_LINE\_1

## ADDRESS\_LINE\_2

### Description

The second line of this address.

## Notes

If you have unit information on a separate line, this may contain the street address.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_ADDRESS\_LINE\_2
- AFTER\_CASS\_ADDRESS\_LINE\_2
- AFTER\_NCOA\_ADDRESS\_LINE\_2

## CITY

### Description

The city for this address.

### Notes

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_CITY
- AFTER\_CASS\_CITY
- AFTER\_NCOA\_CITY

## STATE

### Description

The state for this address.

### Notes

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_STATE
- AFTER\_CASS\_STATE
- AFTER\_NCOA\_STATE

## ZIP\_CODE

### Description

The ZIP Code for this address.

### Notes

During CASS processing, successfully corrected addresses will receive a valid ZIP + 4 Code in this field.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_ZIP\_CODE
- AFTER\_CASS\_ZIP\_CODE
- AFTER\_NCOA\_ZIP\_CODE
- AFTER\_NCOA\_QUERY\_ZIP\_CODE

## LAST\_LINE

### Description

Contains the city, state and ZIP Code for this address.

### Notes

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_LAST\_LINE
- AFTER\_CASS\_LAST\_LINE
- AFTER\_NCOA\_LAST\_LINE

## COUNTY\_NAME

### Description

The name of the county in which this address is located.

### Notes

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_COUNTY\_NAME
- AFTER\_NCOA\_COUNTY\_NAME

## COUNTY\_CODE

### Description

The code for the county in which this address is located.

### Notes

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_COUNTY\_CODE
- AFTER\_NCOA\_COUNTY\_CODE

## URBANIZATION

### Description

The urbanization for this address.

### Notes

This field applies to Puerto Rico addresses only.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_URBANIZATION
- AFTER\_CASS\_URBANIZATION
- AFTER\_NCOA\_URBANIZATION

## COUNTRY

### Description

The country of this address.

## Notes

### CONGRESSIONAL\_DISTRICT

#### Description

Returns the congressional district of this address.

## Notes

Output only.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_CONGRESSIONAL\_DISTRICT
- AFTER\_NCOA\_CONGRESSIONAL\_DISTRICT

### ADDRESS\_BLOCK

#### Description

The full address as would be printed on the front of a mail piece.

## Notes

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_ADDRESS\_BLOCK
- AFTER\_NCOA\_ADDRESS\_BLOCK

### EXTRA\_INFO

#### Description

Contains extra address information that the service was unable to parse into any other fields.

## Notes

Output only.

### CASSDATE

#### Description

An encoded string that contains the date that this record was last processed.

## Notes

Output only.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_CASSDATE
- AFTER\_NCOA\_CASSDATE

## ERROR\_CODE

### Description

A numerical code that indicates the results of CASS processing.

## Notes

Output only.

See the appendix for the codes.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_ERROR\_CODE
- AFTER\_NCOA\_ERROR\_CODE

## ERROR\_STRING

### Description

Text that explains the results from CASS processing.

## Notes

Output only.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_ERROR\_STRING
- AFTER\_NCOA\_ERROR\_STRING

## LOT\_NUMBER

### Description

The line-of-travel number for this address.

### Notes

Output only.

Returned from CASS processing.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_LOT\_NUMBER
- AFTER\_NCOA\_LOT\_NUMBER

## CARRIER\_ROUTE

### Description

The carrier route for this address.

### Notes

Output only.

This information is returned during CASS processing, which is performed during the MOVEService process.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_CARRIER\_ROUTE
- AFTER\_CASS\_CARRIER\_ROUTE
- AFTER\_NCOA\_CARRIER\_ROUTE

## DPC

### Description

Contains the delivery point code for this address.

### Notes

Output only.

This data is generated during CASS processing.

Has the following related fields that show this data before and after various processes:

- BEFORE\_CASS\_DPC
- AFTER\_CASS\_DPC
- AFTER\_NCOA\_DPC

## DP\_BARCODE

### Description

The delivery point barcode for this address record.

### Notes

Output only.

This data will be generated during CASS processing.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_DP\_BARCODE
- AFTER\_NCOA\_DP\_BARCODE

## USER\_DEFINED fields

### Description

Custom data to attach to a record.

### Notes

There are 15 user defined fields for you to use, named USER\_DEFINED\_1 to USER\_DEFINED\_15

## BEFORE\_CASS\_PRIMARY\_NUMBER

### Description

The street number for the address as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_PRIMARY\_NUMBER
- AFTER\_NCOA\_PRIMARY\_NUMBER

## BEFORE\_CASS\_PRE\_DIRECTIONAL

### Description

The pre-directional, if any, as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_PRE\_DIRECTIONAL
- AFTER\_NCOA\_PRE\_DIRECTIONAL

## BEFORE\_CASS\_POST\_DIRECTIONAL

### Description

The post-directional information, if any, as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_POST\_DIRECTIONAL
- AFTER\_NCOA\_POST\_DIRECTIONAL

## BEFORE\_CASS\_STREET\_NAME

### Description

The name of the street as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_STREET\_NAME
- AFTER\_NCOA\_STREET\_NAME

## BEFORE\_CASS\_SUFFIX

### Description

The street suffix as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_SUFFIX
- AFTER\_NCOA\_SUFFIX

## BEFORE\_CASS\_UNIT\_NUMBER

### Description

The unit number as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_UNIT\_NUMBER
- AFTER\_NCOA\_UNIT\_NUMBER

## BEFORE\_CASS\_UNIT\_DESIGNATOR

### Description

The unit type designator as it was before CASS processing.

### Notes

This information is parsed from ADDRESS\_LINE\_1 and ADDRESS\_LINE\_2.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_UNIT\_DESIGNATOR
- AFTER\_NCOA\_UNIT\_DESIGNATOR

## AFTER\_CASS\_PMB\_NUMBER

### Description

The postal mailbox number, if any, of the address after being CASS processed.

### Notes

Has a related field for this data after NCOA processing.

- AFTER\_NCOA\_PMB\_NUMBER

## ZIP4\_FOOTNOTE

### Description

Returns information about the CASS process.

### Notes

Output only.

For explanation of codes, see DPV\_FOOTNOTE

## DPV\_CODED

### Description

Indicates whether an address was verified using DPV.

### Notes

Output only.

For an address to receive a valid ZIP + 4 Code during CASS processing, it must be verified by DPV.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_DPV\_CODED
- AFTER\_NCOA\_DPV\_CODED

## DPV\_IS\_CMRA

### Description

Indicates whether this address is a commercial mail-receiving agent.

### Notes

Output only.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_DPV\_IS\_CMRA
- AFTER\_NCOA\_DPV\_IS\_CMRA

## DPV\_IS\_VACANT

### Description

Indicates whether this address has been vacant for at least 90 days.

### Notes

Output only.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_DPV\_IS\_VACANT
- AFTER\_NCOA\_DPV\_IS\_VACANT

## DPV\_INDICATOR

### Description

Returns a single character that describes the result of DPV processing.

### Notes

Output only.

Returns one of the following:

- Y – Both the primary and secondary address data validated against the DPV database.
- S – The primary address is valid, but the secondary is invalid.
- D – The primary address is valid, but the address is missing secondary information.
- N – The primary address is not valid. This address was not given a ZIP + 4 Code.
- "" – The address was not presented to the DPV table because it was missing components needed for the lookup.
- X – The DPV database has been locked-out because of a protocol violation. You must unlock DPV before any more addresses can be processed with DPV.
- E – The DPV data file is more than 105 days old. By USPS restrictions, no more addresses can be presented to the DPV table.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_DPV\_INDICATOR
- AFTER\_NCOA\_DPV\_INDICATOR

## DPV\_FOOTNOTE

### Description

Indicates the results of the DPV verification process.

### Notes

Output only.

Returns one of the following:

- LK – Processing locked out due to a seed record being processed
- AA – Matched to the ZIP+4 file
- A1 – No match against the ZIP+4 file
- BB – Matched to DPV file (all components confirmed)

- CC – Matched only after removing secondary Information; they were presented but invalid
- N1 – Input Primary matched, but high-rise missing secondary number
- M1 – Primary number missing
- M3 – Primary number invalid
- P1 – Input missing PO, RR, HC box number
- P3 – Failed DPV because of invalid PO, RR, or HC box number
- RR – Matched CMRA (found in CMRA file)
- R1 – Matched CMRA, but secondary number (i.e., PMB) missing
- U1 – Matched unique zip code
- G1 – Matched General delivery
- F1 – Matched military address

## LACS\_CODED

### Description

Indicates if this address was changed due to a match in the LACS database.

### Notes

Output only.

LACS processing occurs as part of CASS processing.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_LACS\_CODED
- AFTER\_NCOA\_LACS\_CODED

## EWS\_CODED

### Description

Indicates that this record was flagged by the early warning system as a new address.

## Notes

Output only.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_EWS\_CODED
- AFTER\_NCOA\_EWS\_CODED

## RECORD\_TYPE

### Description

The type of address record this is.

## Notes

Output only.

Returns one of the following characters:

- S – Street record
- P – Post office box
- R – Rural Route or Highway Contract
- H – High-rise, Building or Apartment
- F – Firm Record
- G – General Delivery
- M – Multi-Carrier Record

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_RECORD\_TYPE
- AFTER\_NCOA\_RECORD\_TYPE

## MATCHED\_TO\_DEFAULT

### Description

This field will return 1 if CASS processing has matched it to a default ZIP Code.

### Notes

Output only.

Default ZIP Codes are for large, single addresses, such as universities or corporations, that do not have +4 codes.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_MATCHED\_TO\_DEFAULT
- AFTER\_NCOA\_MATCHED\_TO\_DEFAULT

## MOVE\_EFFECTIVE

### Description

The date that the matched move became or becomes the record's active address.

### Notes

Output only.

## MOVE\_TYPE

### Description

The type of move found.

### Notes

Output only.

This returns one of the following:

- I – Individual
- F – Family
- B – Business

## MATCH\_FLAG

### Description

Data about the match that the MOVEService made with this address.

### Notes

Output only.

Returns one of the following:

- M – Matched; updated address
- F – Foreign Move; new address unavailable
- K – No forwarding address; new address unavailable
- G – PO box closed; new address unavailable
- N – No match
- X – Other

## MOVE\_FOOTNOTE

### Description

Describes the results of the MOVEService processing.

### Notes

Output only.

This field will contain one of the following:

Match Found – new address returned

- A – Input record matched
- 91 – Secondary number dropped from change of address
- 92 – Secondary number dropped from input address

Match Found – new address unavailable

- 1 – Foreign move
- 2 – Move left no address
- 3 – PO box closed; no forwarding
- 5 – New 11-digit DPBC is ambiguous
- 14 – New address would not convert to deliverable
- 19 – New address not ZIP + 4 coded

#### No Match Found

- 00 – No move found
- 4 – Street address missing secondary
- 6 – Conflicting directions, middle name related
- 7 – Conflicting directions, gender related
- 8 – Other conflicting instructions
- 9 – High-rise default
- 10 – Rural route default
- 11 – Individual, insufficient name for match
- 12 – Middle name test failed
- 13 – Gender test failed
- 15 – Individual name insufficient
- 16 – Secondary number discrepancy
- 17 – Other insufficient name
- 18 – General delivery

- 20 – Conflicting directions after rechaining
- 66 – Address deleted, no forwarding allowed

## MOVE\_FOOTNOTE\_SHORT\_DESCRIPTION

### Description

Returns a short description of the results of the MOVEService processing.

### Notes

Output only.

## MOVE\_FOOTNOTE\_LONG\_DESCRIPTION

### Description

Returns a longer, more detailed description of the results of the MOVEService processing.

### Notes

Output only.

## LACS\_FOOTNOTE

### Description

Provides additional information about the LACS process.

### Notes

Output only.

Returns one of the following:

- blank – Not processed / Seed record.
- 00 – No match.
- 09 – Matched to default high-rise address; address not updated.
- 14 – Match failed to build new address.

- 92 – Match secondary dropped from input.
- A – Match success.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_LACS\_FOOTNOTE
- AFTER\_NCOA\_LACS\_FOOTNOTE

## LACS\_INDICATOR

### Description

Indicates the results of a LACS process.

### Notes

Output only.

Returns one of the following:

- blank – Not processed.
- N – Match, but there was a failure to build new address.
- Y – Match success, new address provided.
- S – Match with secondary dropped from input.
- F – Seed record.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_LACS\_INDICATOR
- AFTER\_NCOA\_LACS\_INDICATOR

## SUITELINK\_FOOTNOTE

### Description

Indicates the results of Suite<sup>Link</sup> matching.

## Notes

Output only.

Returns one of the following:

- blank – Was not processed by the Suite<sup>Link</sup> engine: the address did not qualify for a lookup. Only default high rise addresses qualify for a Suite<sup>Link</sup> lookup.
- A – The address was processed and secondary information was added to the resulting address.
- 00 – The address was processed through the Suite<sup>Link</sup> engine, but did not result in a successful match; no secondary information was added.

Has the following related fields that show this data before and after various processes:

- AFTER\_CASS\_SUITELINK\_FOOTNOTE
- AFTER\_NCOA\_SUITELINK\_FOOTNOTE

# Results Codes

## Address Correction Errors and Results

The error codes in the following table indicate the results of an attempt to correct an address as stored in the ERROR\_CODE field. Codes 0 – 99 are considered successful.

VALUE	DESCRIPTION
<b>Corrected</b>	
0, 7, 8	No Changes
0	No significant changes
7	Matched to unique ZIP code; delivery address line not verified
8	Matched to general delivery ZIP code; delivery address line not verified
9, 10	LACS <sup>Link</sup> / Suite <sup>Link</sup> Changes
9	Address revised by LACS <sup>Link</sup>
10	Secondary information added by Suite <sup>Link</sup>
11-20	Special Situations
11	Alias match
12	ZIP move match
13	Alternate match
21-30	Last Line Changes
21	City changed

22	City and state changed
23	City and ZIP changed
24	City, state, and ZIP changed
25	ZIP changed
26	ZIP and State changed
27	State changed
31-99	Delivery Address Line Changes
31	Street name changed
32	Street name and suffix changed
33	Street name and Directional changed
34	Street name, suffix, and directional changed
35	Street name and city changed
36	Street name, city, and state changed
37	Street name, city, and ZIP changed
38	Street name, city, state, and ZIP changed
39	Street name and ZIP changed
40	Street name, ZIP, and state changed
41	Street name and state changed
42	Street name, suffix, and city changed
43	Street name, suffix, city, and state changed

44	Street name, suffix, city, and ZIP changed
45	Street name, suffix, city, state, and ZIP changed
46	Street name, suffix, and ZIP changed
47	Street name, suffix, ZIP, and state changed
48	Street name, suffix, and state changed
49	Street name, directional, and city changed
50	Street name, directional, city, and state changed
51	Street name, directional, city, and ZIP changed
52	Street name, directional, city, state, and ZIP changed
53	Street name, directional, and ZIP changed
54	Street name, directional, ZIP, and state changed
55	Street name, directional, and state changed
56	Street name, suffix, directional, and city changed
57	Street name, suffix, directional, city, and state changed
58	Street name, suffix, directional, city, and ZIP changed
59	Street name, suffix, directional, city, state, and ZIP changed
60	Street name, suffix, directional, and ZIP changed
61	Street name, suffix, directional, ZIP, and state changed
62	Street name, suffix, directional, and state changed
63	Suffix changed
64	Suffix and directional changed

65	Suffix and city changed
66	Suffix, city, and state changed
67	Suffix, city, and ZIP changed
68	Suffix, city, state, and ZIP changed
69	Suffix and ZIP changed
70	Suffix, ZIP, and state changed
71	Suffix and state changed
72	Suffix, directional, and city changed
73	Suffix, directional, city, and state changed
74	Suffix, directional, city, and ZIP changed
75	Suffix, directional, city, state, and ZIP changed
76	Suffix, directional, and ZIP changed
77	Suffix, directional, ZIP, and state changed
78	Suffix, directional, and state changed
79	Directional changed
80	Directional and city changed
81	Directional, city, and state changed
82	Directional, city, and ZIP changed
83	Directional, city, state, and ZIP changed
84	Directional and ZIP changed
85	Directional, ZIP, and state changed

86	Directional and state changed
91-99	DPV Warnings
92	Failed DPV because of invalid secondary
93	Failed DPV because of missing secondary
<b>Uncorrected</b>	
111-120	Last Line Parsing
111	No input ZIP, no input state, and no input city
112	No input ZIP and no input city
113	Foreign address
211-220	Last Line Retrieval
211	No input ZIP, no input state, and input city invalid
212	No input ZIP, input state invalid, and input city invalid
213	Input ZIP invalid, no input state, and input city invalid
214	Input ZIP invalid, input state invalid, and input city invalid
215	Input ZIP invalid, no input city
216	Unique input ZIP Code does not match input city/state
217	Invalid city, missing or invalid state, and invalid ZIP Code
218	Invalid city and ZIP Code

219	Missing state and ZIP Code
220	Invalid city and missing ZIP Code
311-320	Address Line Parsing
311	Could not parse a primary number from input data
312	Could not parse a street name from input data
313	Address lines blank
411-430	Address Line Retrieval
411	Primary number invalid
412	Street name invalid
413	Similar street names were found but with no exact matches
414	Multiple possible matches with different ZIP+4 codes were found
415	Predirectional required to choose from multiple possible matches
416	Postdirectional required to choose from multiple possible matches
417	Suffix required to choose from multiple possible matches
418	Directional and suffix required to choose from multiple possible matches
419	Valid ZIP Code required to choose from multiple possible matches
420	Valid city name required to choose from multiple possible matches
421	Valid urbanization required to choose from multiple possible matches
422	Matched to undeliverable address; 5-digit coded

423	Matched to a record in the Early Warning System (EWS) file; an exact match will be possible with the next database update
491-499	DPV Failures
491	Failed DPV because of invalid primary
492	Valid primary but failed DPV because of invalid secondary
493	Valid primary but failed DPV because of missing secondary
494	Failed DPV because of invalid PO, RR, or HC box number

## NCOALink Footnote Codes

NCOALink footnotes are returned when the BCC Software Move Update service compares your mailing to the USPS® NCOALink data set. The footnotes provide information about the records processed through the service.

When an address is checked against the NCOALink data set, a new address may or may not be found, and the new address may or may not be provided, based on certain conditions. Each set of conditions has a different code, called a footnote code. The footnote is returned by the service to help you understand the outcome and analyze your records.

The following tables explain Footnote Codes for conditions for cases where:

- A match was found and a new address was provided
- A match was found, but a new address was not available
- A match was found, but the service was unable to provide a new address
- An adequate match was not found, so a new address was not provided

Columns include:

- NCOALink Footnote Code: The code returned by USPS® in the NCOALink process
- Description: An explanation of the outcome
- New Address Provided?: Whether or not the service provided a new address

### A match was found and a new address was provided

NCOALink Footnote Code	Description	New Address Provided?
A	<p><b>Full match.</b></p> <p>The address was matched to an NCOALink record, and a new address has been provided.</p>	Yes
91	<p><b>Matched despite missing secondary number.</b></p> <p>The address was matched to an NCOALink record. The address submitted did not include a secondary number (such as an apartment or suite number), and the NCOALink record did. The secondary number was not used in making the match because the match was close enough.</p> <p>A new address has been provided.</p>	Yes
92	<p><b>Matched despite extra secondary number.</b></p> <p>The address was matched to an NCOALink record. The address submitted included a secondary number (such as an apartment or suite number), and the NCOALink address did not. The secondary number was not used in making the match because the match was close enough.</p> <p>A new address has been provided.</p>	Yes

## A match was found but no new address was available

NCOA Link Footnote Code	Description	New Address Provided?
1	<p><b>New address is outside US</b></p> <p>The address was matched to an NCOA<sup>Link</sup> record, but the new address was outside the USPS<sup>®</sup> delivery area.</p> <p>Because the new address is not served by USPS<sup>®</sup>, a new address could not be given.</p>	No
2	<p><b>No forwarding address.</b></p> <p>The address was matched to an NCOA<sup>Link</sup> record, but the person did not provide their new address to USPS.</p> <p>Because no new address is on record, a new address could not be given.</p>	No
3	<p><b>PO Box closed.</b></p> <p>The address was matched to an NCOA<sup>Link</sup> record, but the submitted address was a PO Box that was closed without a forwarding address.</p> <p>Because no forwarding address is on record, a new address could not be given.</p>	No

## A match was found but the service is unable to provide a new address

NCOA Link Footnote Code	Description	New Address Provided?
5	<p><b>Too many matches for new address.</b></p> <p>The address was matched to an NCOA<sup>Link</sup> record, but the Delivery Point Bar Code (DPBC) on the NCOA<sup>Link</sup> record matches more than one delivery address.</p> <p>Because there is more than one possible address match, a new address could not be given.</p>	No
14	<p><b>Undeliverable address.</b></p> <p>The address was matched to an NCOA<sup>Link</sup> record, but the new address on record is not a deliverable address.</p> <p>Because the new address is not recognized by USPS as a deliverable address, a new address could not be given.</p>	No
19	<p>Either:</p> <ul style="list-style-type: none"> <li>• a) <b>ZIP Code not found.</b>  <p>The address was matched to an NCOA<sup>Link</sup> record, but the new address does not match a ZIP+4 code.</p> <p>Because the ZIP+4 code could not be found for the new address, a new address could not be given.</p> </li> </ul> <p>-or-</p> <ul style="list-style-type: none"> <li>• b) <b>Temporary change of address.</b>  <p>The address was matched to an NCOA<sup>Link</sup> record for a temporary change of address. At the request of the recipient, First Class mail will be forwarded to a temporary address for up to 12 months. Standard mail will not be forwarded.</p> <p>The NCOA<sup>Link</sup> system cannot provide the temporary address.</p> </li> </ul>	No

The service did not find an adequate match

NCOALink Footnote Code	Description	New Address Provided?
0	<p><b>No matching address.</b></p> <p>The address could not be found in the NCOALink database.</p> <p>Because the provided address was not found, a new address could not be given.</p>	No
4	<p><b>Apartment number missing for family move.</b></p> <p>The address was partially matched to an NCOALink record for a family (multiple people at the same address), but there is not enough information to match to a single individual. The NCOALink address on record includes secondary address information (such as an apartment or suite number), but the submitted address does not.</p> <p>Because the apartment number is missing and a match to an individual could not be made, a new address could not be given.</p>	No
6	<p><b>More than one possible address match, but middle names do not match.</b></p> <p>The address was partially matched to NCOALink records, but there is more than one possible match. In addition, the middle names for the individual and the NCOALink records do not match.</p> <p>Because there is more than one possible match and the middle names do not match, a new address could not be given.</p>	No

NCOALink Footnote Code	Description	New Address Provided?
7	<p><b>More than one possible address match, but genders do not match.</b></p> <p>The address was partially matched to NCOALink records, but there is more than one possible match. In addition, the genders for the first names of the individual and the NCOALink records do not match.</p> <p>Because there is more than one possible match and the genders do not match, a new address could not be given.</p>	No
8	<p><b>Too many possible matches.</b></p> <p>The address was matched to more than one NCOALink record. There may be more than one record on file when a change of address is recorded more than once for an individual, a family, and/or a business.</p> <p>Because there is more than one possible match, a new address could not be given.</p>	No
9	<p><b>Family move with high rise address does not match individual name.</b></p> <p>The address was partially matched to an NCOALink record, but the NCOALink record is for a family (multiple people at the same address). Individual names do not match, and the NCOALink record is for a high rise address, where there could be many possible apartments or delivery points.</p> <p>Because there is no exact match to an individual in a building with multiple delivery addresses, a new address could not be given.</p>	No

NCOALink Footnote Code	Description	New Address Provided?
10	<p><b>Family move with default rural address does not match individual name.</b></p> <p>The address was partially matched to an NCOALink record, but the NCOALink record is for a family (multiple people at the same address). Individual names do not match, and the NCOALink record is for a rural address that includes only the default route and not a specific new address.</p> <p>Because there is no exact match to an individual for a general rural address, a new address could not be given.</p>	No
11	<p><b>Only a last name could be matched.</b></p> <p>The address was partially matched to an NCOALink record, but the NCOALink record does not contain first name information to be able to match to an individual.</p> <p>Because there is no first name information available in the NCOALink record, a new address could not be given.</p>	No
12	<p><b>Middle name does not match.</b></p> <p>The address was partially matched to an NCOALink record, but the middle name or initial submitted is different than the NCOALink records.</p> <p>Because the middle name does not match, a new address could not be given.</p>	No
13	<p><b>Gender does not match.</b></p> <p>The address was partially matched to an NCOALink record, but the gender of the name submitted is different than the gender of the NCOALink record.</p> <p>Because the gender does not match, a new address could not be given.</p>	No

NCOALink Footnote Code	Description	New Address Provided?
15	<p><b>Missing first name.</b></p> <p>The address was partially matched to an NCOALink record, but the address submitted does not have a first name or uses only initials.</p> <p>Because the first name of the submitted address is missing, a new address could not be given.</p>	No
16	<p><b>Apartment number does not match.</b></p> <p>The address was partially matched to an NCOALink record, but the secondary information (such as an apartment or suite number) in the submitted address is different than in the NCOALink record.</p> <p>Because the apartment number could not be matched, a new address could not be given.</p>	No
17	<p><b>First name does not match.</b></p> <p>The address was matched to an NCOALink record, but the first name does not match or did not contain enough information to match.</p> <p>Because the first name could not be matched, a new address could not be given.</p>	No
18	<p><b>Family move with General Delivery address does not match individual name.</b></p> <p>The address was partially matched to an NCOALink record, but the NCOALink record is a General Delivery address for a family (multiple people at the same address), and no individual match could be found.</p> <p>Because a General Delivery address could not be matched to an individual, a new address could not be given.</p>	No

NCOALink Foot-note Code	Description	New Address Provided?
20	<p><b>Delivery code could not be confirmed.</b></p> <p>The address was matched to an NCOALink record, but the new address either did not match a ZIP+4 code or could not be confirmed as a deliverable address.</p> <p>Because the new address did not match a delivery code, a new address could not be given.</p>	No
66	<p><b>Deleted address with no forwarding allowed.</b></p> <p>The address was matched to an NCOALink record, but the NCOALink record is scheduled to be deleted and no mail may be forwarded from the address.</p> <p>Because forwarding is not allowed, a new address could not be given.</p>	No

## NCOALink Move Type Codes

The Change of Address record is categorized by the type of move: for a business, for a family, or for an individual. Each move type uses a code:

Code	Description
I	Individual
F	Family
B	Business

## Additional Resources

The following resources are available to help you with your software.

### Knowledge Base

BCC Software offers tips, tricks, and best practices for using our products. Knowledge Base and Support articles can help empower both experts and new users.

- To learn more, visit the [BCC Software Knowledge Base on the BCC Software Customer Portal](#) ⇨.

### How to Contact Support

- BCC Software Technical Support online:  
<https://bccsoftware.com/customer-center/customer-support/> ⇨
- Email: [support@bccsoftware.com](mailto:support@bccsoftware.com) ⇨