Quest® Coexistence Manager™ for Notes 3.8.3

Program Parameters Reference

Contents

About the CMN documentation suite

The documentation for Quest Coexistence Manager for Notes (CMN) includes:

- **Release Notes** (printable PDF): Describes the current CMN release—any new and enhanced features, resolved issues, and known issues. Also documents minimum installation requirements, and provides Quest contact information.

- **Quick-Start Guide** (printable PDF): An orientation to the product's basic purposes, features and capabilities, with a case study showing how its primary components are most commonly used within a typical coexistence scenario. Also explains how to download and install the software.

- **CMN User Guide** (printable PDF): Comprehensive documentation of CMN's three primary components: Directory Connector, Mail Connector and Free/Busy Connector. Describes component capabilities, deployment considerations, configuration instructions and tips, and application notes and screen-by-screen field notes for CMN's Management Console software tools.

- **FBC Scenarios Guide** (printable PDF): Provides process instructions and application notes for installing and configuring CMN's Free/Busy Connector (FBC) in a variety of Exchange-side scenarios.

- **CMN Program Parameters Reference** (printable PDF): Listing of all CMN program parameters that are not associated with UI fields in CMN's Management Console, with descriptions and default values and usage/application notes. (Parameters associated with UI fields do appear in the *Configuration.xml* files, but should not be edited manually.)

- **Management Console Online Help** (three compiled Windows Help files, one for each CMN component): Field notes and application notes for the screens and features of CMN's Management Console.

All CMN documentation is intended for network administrators, consultants, analysts, and any other IT professionals who will install or use the product components, or who may help plan for their use in a coexistence scenario. All of these documents, including the online Help, are bundled and installed with the product, and all except the Help files are also available separately at Quest's Support Portal.

Where To Look in the CMN Documentation

This table shows where you can find particular types of information about particular CMN components:

	for Dir Connector & Mail Connector	for Free/Busy Connector
Introduction and orientation:	— — CMN Quick-Start Guide and User Guide — —	
System requirements:	— — CMN Release Notes — —	
Installation instructions:	— — CMN Quick-Start Guide — —	
Configuration instructions:	CMN User Guide	CMN FBC Scenarios Guide
Operating instructions:	— — CMN User Guide — —	
Troubleshooting info:	CMN User Guide	CMN FBC Scenarios Guide

The CMN application Help files contain the same information as the *User Guide*, but make the information available on-screen at a single keystroke (from the CMN Management Console).

All CMN documentation is intended for network administrators, consultants, analysts, and any other IT professionals who will install or use the product components, or who may help plan for their use in a coexistence scenario. All of these documents, including the online Help, are bundled and installed with the product, and all except the Help files are also available separately at Quest's Support Portal.

About CMN Program Parameters

- Introduction to CMN parameters
- Parameter format and structure
- How to specify or edit program parameters

Introduction to CMN parameters

CMN's Directory Connector, Mail Connector and Free/Busy Connector are configurable, and most of their feature options are set in CMN's Management Console. The Management Console is a friendly interface between you and a set of configuration files that CMN components read every time they are started.

Configuration settings for CMN's different components are saved in separate configuration files: a *Configuration.xml* file for each of the Directory Connector and Mail Connector (two files with the same name, but saved in the different CMN component subfolders), and another set of configuration files for the Free/Busy Connector (in the FBC's component subfolder). The Management Console provides screens with labeled fields to simplify entering and editing the configuration settings in those files.

When you start the Management Console, CMN automatically loads the currently active configuration values for all three CMN components. Once the Management Console is running, however, component configuration data is saved and opened separately, for one component at a time, as shown here:
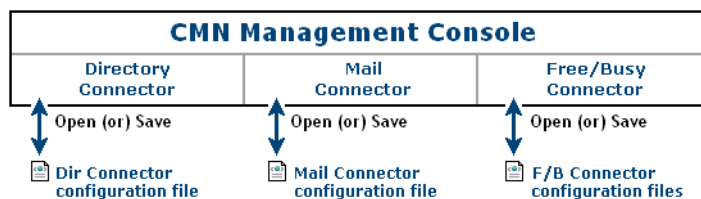
While the Management Console lets you enter and change most CMN feature options, some options can be set and changed only by manually editing the corresponding *Configuration.xml* file. This *Program Parameters Reference* documents those parameters.

> **i** | **NOTE:** All FBC program parameters in the current release are associated with UI fields in the FBC Management Console. As such, these parameters are not documented in this *Reference*, since they should not be edited manually (in the xml files) except at the direction of Quest Support.

The contents of a configuration XML file can be edited by any XML editor, as described separately below.

Parameter format and structure

CMN parameters are specified and stored in XML Unicode text files that conform to XML markup standards and conventions. Each parameter is specified by an XML element, consisting of a start tag, an end tag, and the parameter value between the two tags. For example:

        <ServicePort>1234</ServicePort>

As with all XML markup tags (and as shown in these examples), the start and end tags are both enclosed within angle brackets, and the end tag is denoted by a slash character ("/") inside the angle brackets preceding the parameter name.

Within a CMN parameters XML file, related parameters are grouped into sections, which also are named and marked by pairs of tags. For example:

```
<ConnectorSettingsList>

    <ConnectorSetting>

        <TargetPort>389</TargetPort>
        <TargetUseSsl>false</TargetUseSsl>
        <SourceOu>DC=mustang,DC=sitraka,DC=com</SourceOu>
        <TargetOu>OU=abcdefg,O=sitraka</TargetOu>
        <Groups>true</Groups>
        <Users>false</Users>
        <Contacts>false</Contacts>
        <SourceGroupObject>msExchDynamicDistribList</SourceGroupObject>

    </ConnectorSetting>

    <AnotherSection>

        <Parameter241>blue</Parameter241>
        <Parameter242>654321</Parameter242>

        <SomeOtherSection>

            <Parameter243>true</Parameter243>
            <Parameter244>kangaroo</Parameter244>
            <Parameter245>Casablanca</Parameter245>

        </SomeOtherSection>

    </AnotherSection>

    <Parameter246>false</Parameter246>

</ConnectorSettingsList>
```

The section tags and parameter elements within an XML file are typically broken into separate lines and indented, as shown here, just for human readability. But the line breaks, indentions and any other white space that occurs between elements are ignored by the programs that read the files.

Spaces that occur within a parameter *value*, however, *are* significant. For example, the spacing within a text string such as:

```
<StreetAddress>123 45th Avenue</StreetAddress>
```

... will be read, interpreted and used by the program in exactly that same form, with spaces intact.

Likewise, upper/lowercase within a parameter value *may* be significant, but only if case matters in whatever context the program will use the string. For example, if *<CityName>baLtImOre</CityName>* and the program will copy that parameter value into a program log or report, or into the user interface as a display element, the name will appear there in that same peculiar form: as "*baLtImOre*." Section and parameter names (tag contents), however, are case-***in***sensitive.

Spelling of parameter names and parameter values is always critical. If a program is looking for a parameter name *UserAddress*, it will not recognize a parameter name *UsreAddress* or *UserAdress*.

Parameter defaults

The default value of a parameter is the value that will prevail in the program if the parameter is omitted from its *Configuration.xml* file.

Parameter value data types

For any given parameter, the parameter value will be meaningful only if it conforms to the parameter's particular data type. Every parameter value is one of these five data types:

- **String:** An alphanumeric string.

- **Integer:** A non-decimal numeric value.

- **Boolean:** A value of "true" or "false" to specify whether a particular feature or function is to be enabled or disabled, or whether a particular operating condition is true or false.

- **Date:** A date, formed by eight digits with two slash characters, in the form: mm/dd/yyyy.

- **Keyword:** An alphanumeric string that matches one of a finite set of valid keyword strings defined for the particular parameter.

Parameters associated with UI fields in CMN's Management Console

Many parameters that appear in the CMN xml configuration files simply store values that are entered or selected in the fields of various screens in CMN's Management Console software. These UI-associated parameters are therefore not documented in this *Reference*, since their values should not be manually edited in the xml files except at the direction of Quest Support.

How to specify or edit program parameters

> ℹ️ **IMPORTANT:** The CMN Management Console must be closed before you open a configuration file as described here. When you select **File|Save** in the Management Console, the Console saves all of its open data to the *Configuration.xml* file, and this will overwrite any manual changes to the file that may have been entered and saved while the Console was open.

> ❗ **CAUTION: Be very careful when manually entering or editing values in the *Configuration.xml* file, where incorrect syntax or even a single-character typo could disable a connector or the DC service altogether.**

The contents of a configuration XML file can be edited by any XML editor:

1 Use an XML editor to open CMN's active XML configuration file. Be sure to open the correct XML file, for the CMN component whose parameter(s) you want to enter or change:

Configuration.XML file for Directory Connector is in:	CMN's \Directory Connector\ folder
Configuration.XML file for Mail Connector is in:	CMN's \Mail Connector\ folder

The configuration files for the Free/Busy Connector are not listed here because all FBC program parameters in the current release are associated with UI fields in the FBC Management Console. As such, these parameters should not be edited manually (in the xml files) except at the direction of Quest Support.

2 Find the parameter within the file. You can text-search for the string of the parameter name, or for the <Section> where the parameter resides.

If the parameter is not already in the file, add it in a new line. In this case, be sure that what you enter matches exactly the parameter name documented in this *Reference*.

3 Enter or change the value of the parameter. Make sure the value is valid by value type, range (if appropriate), and syntax as documented in this *Reference*.

4 Save and close the *Configuration.XML* file.

Directory Connector parameters

> **NOTE:** Parameters that appear in the *DC Configuration.xml* file but that are not listed here are associated with UI entries in the DC Management Console, and should not be edited manually (in the xml file) except at the direction of Quest Support.

<AppendAttributes>

Type: string. *Default:* [null] (connector will overwrite target attributes with source attributes if they differ)

Specifies a list of one or more target object attributes whose values will be *appended to*, rather than *replaced by* source attribute data, when the connector is running in *Update* mode and attribute data differs between an existing object in the target and a corresponding source object. Available for either N-to-E or E-to-N connector, specified separately per connector.

This parameter is optional, and is configured separately for each DC connector in the connector's section of the DC *Configuration.xml* file. To designate the attribute(s) that should function this way in *Update* mode for a particular connector, find (or add) the *<AppendAttributes>* tags for the connector, and enter the names of the target attributes separated by a pipe character ("|") between the tags, like this:

    <AppendAttributes>cn|mail|proxyaddresses</AppendAttributes>

Note that a manual change to an attribute value in a CMN target container will persist through subsequent connector runs if the corresponding source object record has not also been changed. A connector compares the source object data to a copy of the source object record from the preceding connector run, and updates the target object only if the source object has changed.

<IgnoreWizardOuCheck>

Type: boolean. *Default:* true (on)

Determines whether the Connector Creation Wizard will (*false*) or will not (*true*) verify data on pages searching for and displaying OUs (insofar as verifications are possible by other entries and/or environmental inferences). Applies to the Directory Connector as a whole (not just a particular connector). Note that the parameter is defined (named) in terms of *not* verifying, so a *true* setting turns *off* the verifications.

<QueryOUTimeout>

Type: integer. **Default:** 0

When set to the default value 0, CMN times out after 5 minutes. You can set this parameter to the number of minutes desired before time out occurs.

<MailAlertDestServer>

Type: string. **Default:** [null]

Specifies a name or IP address for the mail server to receive a Mail alert report.

This parameter is optional.

<MailAlertFrom>

Type: string. **Default:** administrator@noreply

Specifies the email address of the sender when the mail alert module is enabled for error sync.

<MailAlertTo>

Type: string. **Default:** [null]

Specifies one or more recipient's email addresses. These values are used by the mail alert module for error sync.

This parameter is optional. Find (or add) the <MailAlertTo> tags in the root section of the DC Configuration.xml file and enter the names of the email addresses separated by a semicolon character (";") between the tags. For example:

```
<MailAlertTo>username1@contoso.com;username2@contoso.com</MailAlertTo>
```

> **i** | **NOTE:** Error Sync
>
> To enable the mail alert module for error sync, you must set values for the following three parameters at the same time:
> - <MailAlertFrom>
> - <MailAlertTo>
> - <MailAlertDestServer>

<ServicePort>

Type: integer. **Default:** 9081

Specifies the communications port for the DC service. Available for the Directory Connector service overall (not specified separately per connector).

Specifies which communications port the Directory Connector should use for data transmissions between the DC service and the DC Management Console, on the server where these applications reside. The port for this is set to 9081 by default, and this will be the correct setting in almost all environments.

But if port 9081 is assigned to some other service or function in your environment, you must change this communications port setting in the Directory Connector configuration file. The *<ServicePort>* parameter is usually at or near the end of the *<DirectoryConnectorSettings>* section.

<SourceContactObject>

Type: string. **Default:** Contacts

Specifies an alternate object type (instead of *Contact* objects) in the source whose objects a connector should sync to the target—to a target object type specified by the *<TargetContactObject>* parameter described below. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *Contact* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<SourceGroupObject>

Type: string. **Default:** Groups

Specifies an alternate object type (instead of *Group* objects) in the source whose objects a connector should sync to the target—to a target object type specified by the *<TargetGroupObject>* parameter described below. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *Group* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<SourceResourceObject>

Type: string. *Default:* Resources

Specifies an alternate object type (instead of *Resource* objects) in the source whose objects a connector should sync to the target—to a target object type specified by the *<TargetResourceObject>* parameter described below. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *Resource* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<SourceUserObject>

Type: string. *Default:* Users

Specifies an alternate object type (instead of *User* objects) in the source whose objects a connector should sync to the target—to a target object type specified by the *<TargetUserObject>* parameter described below. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *User* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<TargetContactObject>

Type: string. *Default:* Contacts

Specifies an alternate object type (instead of *Contact* objects) in the target to be synchronized with objects from the source of the type specified by the *<SourceContactObject>* parameter described above. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *Contact* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<TargetGroupObject>

Type: string. *Default:* Groups

Specifies an alternate object type (instead of *Group* objects) in the target to be synchronized with objects from the source of the type specified by the *<SourceGroupObject>* parameter described above. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *Group* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<TargetResourceObject>

Type: string. *Default:* Resources

Specifies an alternate object type (instead of *Resource* objects) in the target to be synchronized with objects from the source of the type specified by the *<SourceResourceObject>* parameter described above. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *Resource* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

<TargetUserObject>

Type: string. *Default:* Users

Specifies an alternate object type (instead of *User* objects) in the target to be synchronized with objects from the source of the type specified by the *<SourceUserObject>* parameter described above. This feature permits the designation of a particular connector to synchronize a particular object type other than the default *User* object. Available for either a N-to-E or E-to-N connector, specified separately per connector, in the connector's <Section> of the *Configuration.xml* file.

Mail Connector parameters

i | **NOTE:** Only a very few of the parameters that appear in the MC *Configuration.xml* file are documented here, because the others are associated with UI entries in the MC Management Console, and should not be edited manually (in the xml file) except at the direction of Quest Support.

<DisableProcessing>

Type: boolean. *Default:* false

Summary: Optional "pass-through" mode to disengage message processing.

Determines whether CMN's optional "pass-through" mode will disengage CMN's message-processing features while CMN continues to pass mail between the servers. This feature is off (*false*) by default, which tells the MC to ***not*** enable its pass-through mode—which would *dis*able message processing. (A *false* setting here tells the MC to *continue* processing messages in addition to routing them to the other environment.) This feature is available only by manually setting this parameter in MC's *Configuration.xml* (is not offered in the Management Console).

<DisableUpdateFromHeader>

Type: boolean. *Default:* false.

Summary: Optionally skip updating from header of the Mime Body in REPLY and COUNTER calendars.

This feature is off (false) by default, which tells the MC to update from header of the Mime Body in REPLY and COUNTER calendars. This feature is available only by manually setting this parameter in MC's `Configuration.xml` file. It cannot be set in the Management Console.

<ForeignSmtpDomain>

Type: domain name *Default:* none

Summary: The SMTP domain name used to route mail from the source to the target server.

<RouterTimeoutSec>

Type: integer. *Default:* 300 (seconds; = 5 minutes)

Summary: Duration of wait for a socket connection.

Sets the number of seconds CMN will wait for a socket connection before timing out.

<RouterWaitSec>

Type: integer. *Default:* 5

Summary: Configure socket channel usage.

This is a "throttling" feature to enforce a time interval for CMN to conclude its communications protocol with a channel before opening another. The actual delay, in seconds, is a "delay factor" (5 by default) x the ratio of the number of open channels to the maximum allowed number of channels. This parameter, in the *<root>* node of the MC's *Configuration.xml*, specifies the delay factor. By default, the Mail Connector waits up to 5 seconds for an available socket channel before opening a new one (if none are then available).

About us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit https://www.quest.com/company/contact-us.aspx or call +1-949-754-8000.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.

Third-party contributions

This product contains the following third-party components. For third-party license information, go to https://www.quest.com/legal/license-agreements.aspx. Source code for components marked with an asterisk (*) is available at https://opensource.quest.com.

Table 1. List of third-party contributions

Component	License or acknowledgment

Index

A

AppendAttributes parameter, 8

C

CMN Mail Connector
    Pass-Through Mode, 11
communications port, Directory Connector, 9
configuration XML file, editing, 7
Configuration.xml files, 5

D

data types, 7
Directory Connector
    communications port, 9
DisableProcessing parameter, 11
DisableUpdateFromHeader, 11

E

editing configuration XML files, 7

F

ForeignSmtpDomain, 11

I

IgnoreWizardOuCheck parameter, 8

M

Mail Connector
    Pass-Through Mode, 11
Management Console fields, parameters for, 5, 7, 8, 11

P

parameter defaults, 6
parameter format and structure, 5
parameter value data types, 7
parameters for Management Console fields, 5, 7, 8, 11
parameters for UI fields, 5, 7, 8, 11
program parameters, editing, 7

Q

QueryOUTimeout, 8

R

RouterTimeoutSec parameter, 11
RouterWaitSec parameter, 11

S

ServicePort parameter, 9
SourceContactObject parameter, 9
SourceGroupObject parameter, 9
SourceResourceObject parameter, 10
SourceUserObject parameter, 10

T

TargetContactObject parameter, 10
TargetGroupObject parameter, 10
TargetResourceObject parameter, 10
TargetUserObject parameter, 10

U

UI fields, parameters for, 5, 7, 8, 11

X

XML file, editing, 7