

---

# Server-side Synchronization in Dynamics CRM and Dynamics CRM Online

**APPLIES TO:** Microsoft Dynamics CRM 2015, Microsoft Dynamics CRM 2016,  
and Microsoft Dynamics CRM Online 2016 Update

**VERSION:** 8.1

**COMPANY:** Microsoft

**RELEASED:** August 2016

---

---

## Copyright

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2016 Microsoft. All rights reserved.

Microsoft, Active Directory, Dynamics, Microsoft Dynamics, Outlook, Windows, Windows PowerShell, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

---

# Table of Contents

|  |           |
|--|-----------|
| <b>Overview</b> .....  | <b>4</b>  |
| <b>Architecture high-level overview</b> .....  | <b>7</b>  |
| Asynchronous Processing Service.....   | 7         |
| Exchange to Dynamics CRM entity mapping.....   | 7         |
| PrepareSync overview.....  | 8         |
| <b>Server roles overview</b> .....   | <b>8</b>  |
| <b>Authentication</b> .....  | <b>9</b>  |
| <b>Tracking email overview</b> .....   | <b>10</b> |
| Outgoing email: Email synchronized from Dynamics CRM to Exchange .....                         | 10        |
| Incoming emails: Email synchronized from Exchange to Dynamics CRM.....                         | 11        |
| Queuing mechanisms, steps, calls to EWS, data merge, threading, and I/O ports .....            | 12        |
| <b>Tracking Appointments, Contacts, and Tasks</b> .....  | <b>14</b> |
| Steps required for queuing mechanisms, calls to EWS, data merge, threading, and I/O ports..... | 15        |
| Sync filters and server-side sync impact.....  | 16        |
| <b>Supportability</b> .....  | <b>16</b> |
| Errors fields in the MailboxBase table .....   | 16        |
| Performance table .....  | 17        |
| Tracing.....   | 18        |
| TraceLogBase table .....   | 18        |
| ETW Tracing.....   | 19        |

---

## Overview

Server-side synchronization, also known as Server-Side sync or Exchange sync, is a server-side process for synchronizing appointments, contacts, tasks (ACTs), and email messages between Exchange Server and Microsoft Dynamics CRM Server. Server-side sync runs as part of the Asynchronous Processing Service. Unlike Outlook synchronization, which requires CRM for Outlook to support synchronization, server-side sync can support synchronizing activities between Dynamics CRM and Exchange without running CRM for Outlook.

Since the server-side sync component is hosted on the server running the Microsoft Dynamics CRM Asynchronous Service server role, it brings some advantages. First, the current Microsoft Dynamics CRM Asynchronous Service already loads the full set of organization metadata-caches. If server-side sync ran in another process, these large caches would have to be loaded a second time resulting in sub-optimal memory use. Second, it gives server-side sync access to the full set of organization metadata-caches that's loaded in the Microsoft Dynamics CRM Asynchronous Service process.

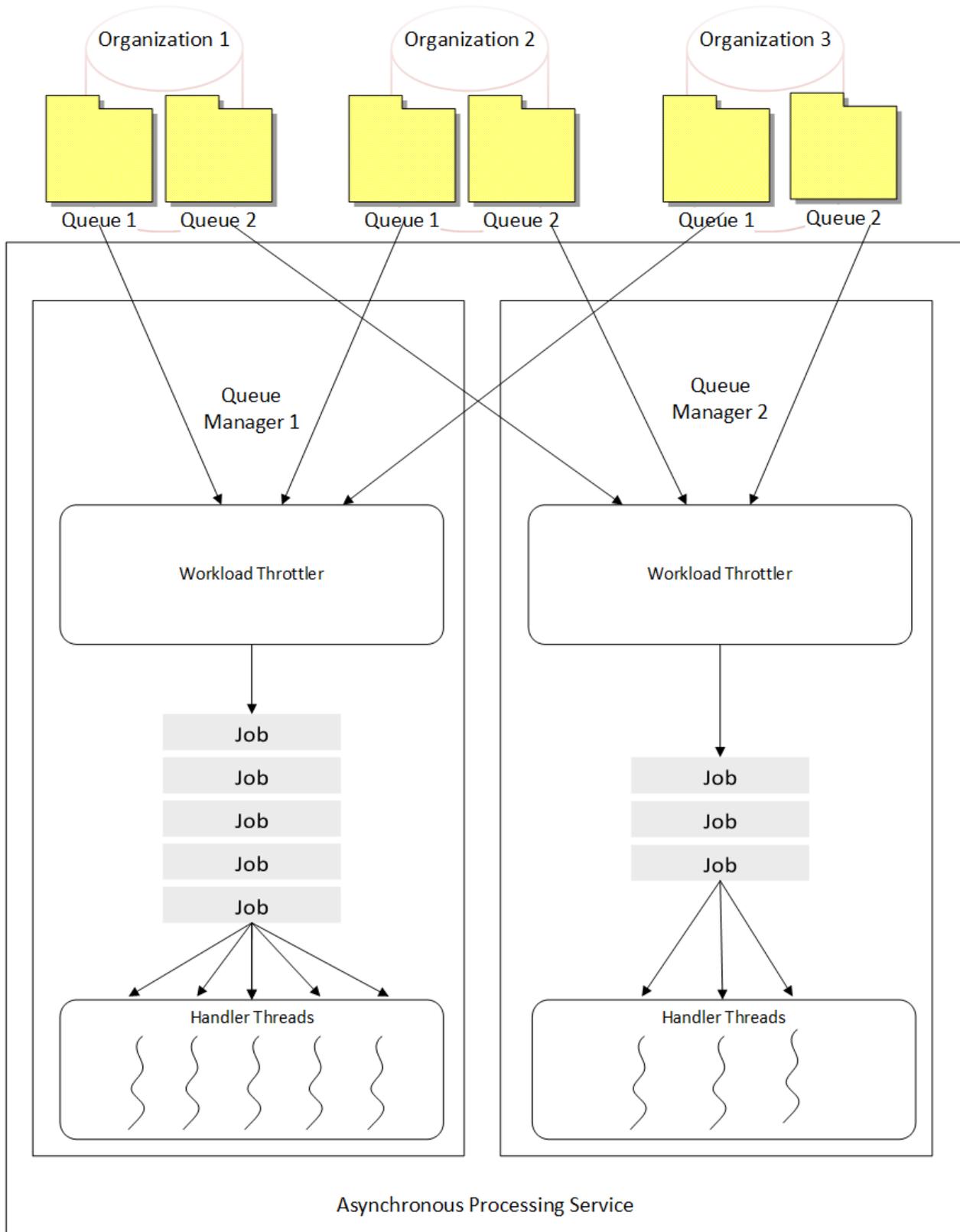
Server-side sync uses a polling mechanism that is mainly operated using mailbox queue and activity queue managers. Both the mailbox and activity queue managers are monitored by the Microsoft Dynamics CRM Asynchronous Service:

- MailboxQueueManager is responsible for picking up mailboxes that are due to be processed and hands them over to appropriate handlers.
- The activity queue picks up the batches of records from the ActivityPointerBase table for all the emails stored in Dynamics CRM Server that are waiting to be sent.

In order to throttle the correct load, server-side sync should poll; the polling frequency is adapted based on factors, such as upper bound limit, existing load, and how active the mailboxes are.

The illustration below provides a high-level workflow of the server-side sync architecture. Each queue-manager holds a timer. When it fires, server-side sync polls mailboxes for email messages from Dynamics CRM into the MailboxQueue or ActivityQueue that are queued for processing in the sync pipeline. Typically, the server-side sync loading mechanism makes sure each mailbox that needs processing is serviced within 15 minutes.

## Server Side Synchronization Architecture



When queuing mailboxes to process, server-side sync provides some configurable values located in the DeploymentProperties table of the configuration database that you can adjust by using Windows PowerShell to customize the queueing capacity in your environment. Notice that this configuration is available for on-premises deployments of Dynamics CRM only. As mentioned before, the polling mechanism works at certain intervals and capacities, which can be controlled by changing the following settings.

| Setting                        | Description   | Default | Lower limit | Upper limit |
|--------------------------------|---|---------|-------------|-------------|
| MailboxQueueSelectInterval     | Controls the frequency of queuing mailboxes.  | 10      | 1           | 60          |
| MailboxesDequedPerEvent        | Controls how many mailboxes are actually queuing for a single event.  | 1       | 1           | 50          |
| MailboxQueueItemsInMemoryHigh  | Defines the upper limit of the number of mailbox batches held in memory at any point in time.   | 350     | 200         | 10000       |
| MailboxQueueItemsInMemoryLow   | Defines the limit of the number of mailbox batches held in memory, beyond which the service will stop removing more mailboxes from the queue. | 100     | 0           | 10000       |
| ActivityQueueItemsInMemoryHigh | Defines the upper limit of the number of activity batches held in memory at any point of time.  | 350     | 200         | 10000       |
| ActivityQueueItemsInMemoryLow  | Defines the lower limit beyond which the service will stop removing activity batches held in memory from the queue.                           | 100     | 0           | 10000       |

The default in memory queue settings are configured for mid-size organizations, which typically work best for organizations that have between 3,000-5,000 users. Queue performance depends on the number of users and item workloads across the servers running the Asynchronous Processing role. You can use Windows PowerShell to either increase or decrease the settings, depending on the number of users and email or activity synchronization experienced. For example, a larger organization can increase the in memory queue item settings, by running these commands:

```
Add-PSSnapin Microsoft.Crm.PowerShell
```

```
#Verify existing server-side sync settings
```

```
Get-CrmSetting -SettingType ServerSideSyncQueueSettings

#Set the values

$set = Get-CrmSetting -SettingType ServerSideSyncQueueSettings

$set.MailboxQueueItemsInMemoryHigh = "500"

$set.MailboxQueueItemsInMemoryLow = "200"

$set.ActivityQueueItemsInMemoryHigh = "500"

$set.ActivityQueueItemsInMemoryLow = "200"

#Commit the settings

Set-CrmSetting -Setting $set

#Verify the new settings

Get-CrmSetting -SettingType ServerSideSyncQueueSettings
```

## Architecture high-level overview

### ***Asynchronous Processing Service***

Server-side sync has a mailbox queue which periodically picks up eligible mailboxes from the organization database and queues them up for processing. The size of the queue, the periodic interval, and the number of mailboxes picked up are all PowerShell configurable values that are adjusted according to the scale of the Asynchronous Processing Service server role deployment. Once the mailboxes are loaded up in the queue they are then worked on by the Workload Throttler, which produces worker threads where each thread picks up a batch of mailboxes for processing. A batch of five mailboxes are picked up per thread as server-side sync has a re-entrant loop mechanism of mailbox processing where the thread gives up the processing context of a mailbox when it makes an asynchronous request to Exchange for data. During this time the next mailbox is picked up and processed. The Workload Throttler throttles the mailbox processing on the basis of certain preconfigured settings. Some of these preconfigured settings have been explained in previously. When the Asynchronous Processing Service server role is deployed on more than one Windows Server there is no affinity between mailboxes and the servers on which they will be processed.

### ***Exchange to Dynamics CRM entity mapping***

The ExchangeSyncIdMapping table in the organization database is primarily designed to work as a cache to store the CrmId and the corresponding Exchange Id's of the items which are synchronized between Exchange and Dynamics CRM. When an activity item is synchronized, its information is stored in the ExchangeSyncIdMapping table and the subsequent seeks for the CrmId for the Exchange item or ExchangeID for the Dynamics CRM item are resolved using this table, thus preventing additional searches. As this table stores information for each

---

synchronized item, it is also used to store any per item error information. Additionally, it's used to store retry and user decision information for errors that require user intervention. For user on-boarding scenarios this table is wiped clean, which forces a full synchronization.

The ExchangeSyncIdMapping table has a direct correlation with the Activity Pointer, Accounts, and Contacts tables because the ExchangeSyncIdMapping table has an entry for each synchronized item and each synchronized item will have a corresponding entry in one of the aforementioned Dynamics CRM activity object tables.

## ***PrepareSync overview***

PrepareSync is the mechanism server-side sync uses to fetch data from Dynamics CRM that needs to be synchronized to Exchange. This mechanism was initially designed to handle user/subscriber initiated synchronizations, such as CRM for Outlook synchronization and go offline synchronization. PrepareSync was later extended to support Exchange synchronization as well. Subscriber information is stored in Subscription and SubscriptionClient tables in the organization database, which also store the last synchronization time stamp in the form of the database version number (CompletedSyncVersionNumber).

PrepareSync involves the following steps:

1. In the first step, a complete scan of the respective Contact and ActivityPointer entity tables is done to find the data that has been modified via create, update, or delete (CRUD) operations, since the last synchronization time. This is stored in the CompletedSyncVersionNumber column of the Subscription table in the organization database. The Id's of the changed items are stored in a table in the tempdb system database.
2. In the second step, per user filters are applied to the data stored in the temp table from the previous step to generate the synchronization information for the current user. This filtered information is then persisted in the SyncEntry<subscriptionid> table of the organization database and is used by the Exchange synchronization or offline synchronization engines, respectively.

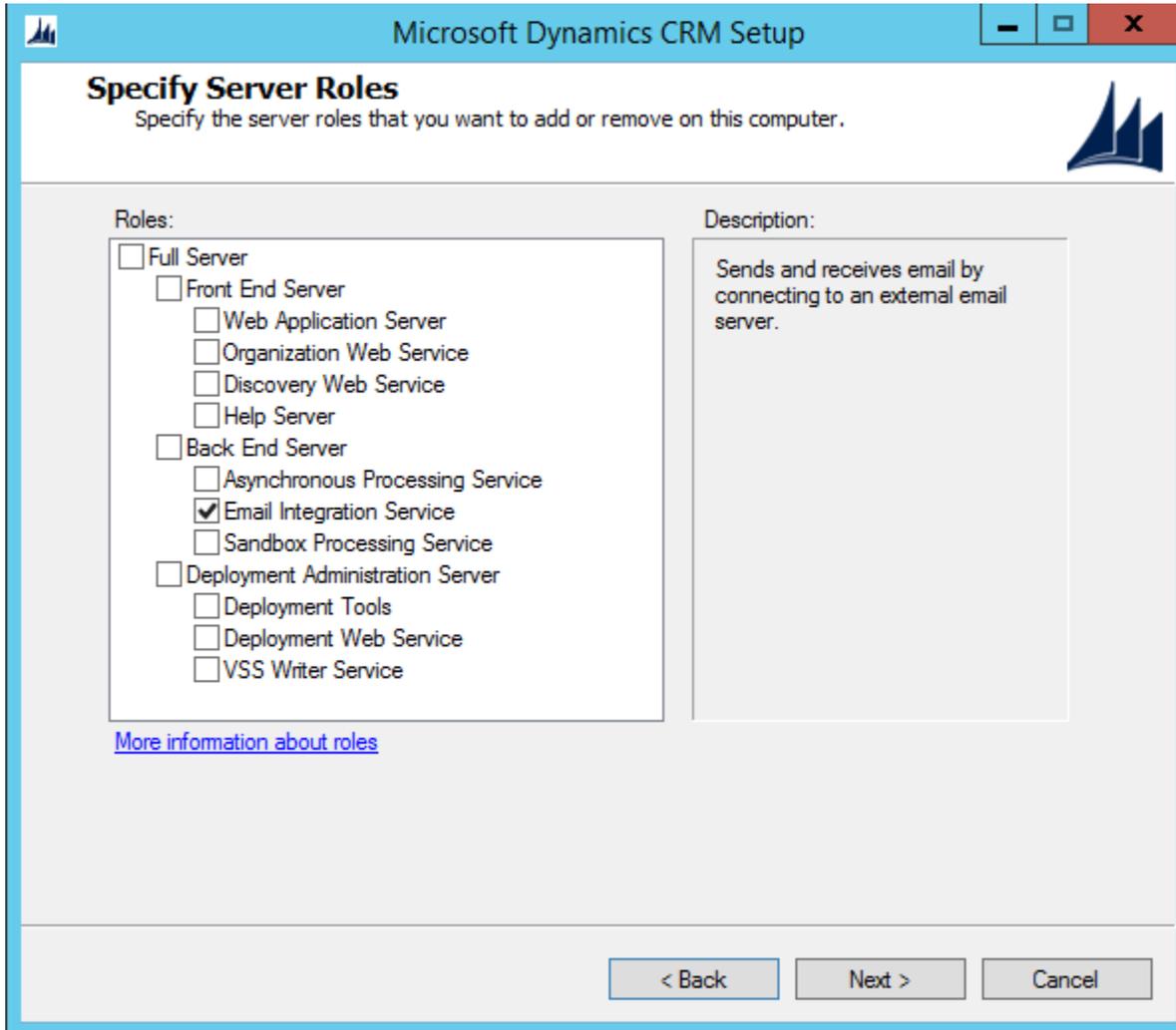
Detailed information about PrepareSync can be found in the [Offline and Online Synchronization in Microsoft Dynamics CRM](#) white paper. Although the paper is written specifically for Microsoft Dynamics CRM 4.0, much of the information still applies to Dynamics CRM 2015 and Dynamics 2016.

## **Server roles overview**

One of the primary goals of server-side sync is to provide the capability where an instance of Dynamics CRM Server can have built-in email integration. With Microsoft Dynamics CRM Online all Dynamics CRM instances include server-side sync capability. With Microsoft Dynamics CRM on-premises deployments, administrators can choose whether to include server-side sync to be included with a deployment of the Asynchronous Processing Service server role.

Dynamics CRM Server Setup facilitates administrators by allowing the selection the of Email Integration Service capabilities as a separate server role. This can help improve performance and scalability by isolating email

integration specific operations for on-premises deployments during installation. The server role can be isolated by selecting only Email Integration Service as indicated in this Dynamics CRM Server Setup wizard screenshot.



## Authentication

One of the first steps to successfully run server-side sync involves configuring an email server profile. In Dynamics CRM on-premises, the email server profile facilitates administrators to specify configuration settings such as server types, server locations, and authentication details.

In order for Dynamics CRM to synchronize emails and ACTs, it has to successfully authenticate with the email server. The "Authenticate Using" option in the email server profile allows various mechanisms to establish this authentication:

- 1. Credentials specified by a user or queue**

This option indicates that every mailbox to be synchronized should specify its own credentials. Those credentials, such as the mailbox user name and password, are used to establish authentication between Dynamics CRM and the email server.

---

## 2. Credentials specified in an email server profile

The email server profile provides an option to specify a common set of credentials that can be specified at the profile level, and can then be used by all mailboxes in the system. This is a convenient means through which an administrator can easily configure and setup the system, and which does not require individual mailboxes to deal with the hassle of providing credentials.

In order for synchronization to work for all mailboxes, it is important to realize that if credentials are specified at the profile level, then that identity should have the appropriate privileges. For instance, the identity may need to have a “Send As” privilege in order to send email on behalf of the mailbox users. Alternatively, the identity may have an even stronger privilege such as impersonation which enables that identity to impersonate users on the server and to perform tasks on their behalf.

When choosing the option to authenticate with credentials at the server profile level, an option has therefore been provided to choose whether to use impersonation or not.

## 3. Windows Integrated Authentication

This option is only available in on-premises deployments. For example, a designated Active Directory user account that is the identity for the Asynchronous Processing Service and is used for authentication.

## 4. Without credentials (anonymous)

Some email servers, such as Exchange, provide the option to configure anonymous access. Therefore, this option is available in the email server profile for on-premises deployments.

Notice that, when you configure Microsoft Dynamics CRM Online with Exchange Online, the email server profile only allows server-to-server authentication and Exchange Autodiscover.

# Tracking email overview

With server-side sync, once the user has configured a mailbox, the emails from the user's Exchange account and their Dynamics CRM account can be kept in sync. This synchronization is done by the Asynchronous Processing Service reading the settings from the organization database.

This involves two main mechanisms:

1. Outgoing emails: To synchronize email from Dynamics CRM Server to a user's Exchange account.
2. Incoming emails: To synchronize email in a user's Exchange account to Dynamics CRM Server.

## Outgoing email: Email synchronized from Dynamics CRM to Exchange

All the email that is created in Dynamics CRM Server is created in the ActivityPointerBase table of the organization database. While processing such email for an organization, a batch of the eligible email from the table are selected and are provided to a WorkLoadThrottler, which then follows the following basic stages:

1. Find email that needs to be synchronized to Exchange, and the related activity parties in Dynamics CRM, as well as find the related attachments.

- 
2. Read the Mailbox configuration to figure out how to connect to the Exchange service. Use Autodiscover if required and get the Exchange Web Services (EWS) endpoint URL.
  3. CreateItemStep: Create an email object in Exchange for each email to be sent.
  4. AddDelegateStep: Add a delegate for each email. This will allow the Dynamics CRM account to send email on behalf of the user.
  5. CreateAttachmentStep: If the email has attachments, create an attachment in Exchange.
  6. SendItemStep: Send the Exchange email to the required activity party recipients.

## Incoming emails: Email synchronized from Exchange to Dynamics CRM

There are two basic mechanisms in which Exchange email can be synchronized to Dynamics CRM:

A. Automated tracking of email:

In this method, the user specific configurations are made once to specify the Exchange server, user's email profile, folders to track, and so on. After these settings are made, the Asynchronous Processing Service continuously polls each mailbox in the organization to find the email that needs to be synchronized to Dynamics CRM. The polling frequency for each mailbox is configurable and is also based on the number of emails that get processed for that mailbox. In this mechanism, each mailbox goes through the following steps:

1. GetInboxFolderHierarchyStep:  
This step gets the folder hierarchy within the Inbox of an Exchange account. This hierarchy gets saved in the MailboxBase table in the organization database.
2. FindItemsStep:  
This step is responsible for finding email in Exchange that need to be synchronized to Dynamics CRM Server, since the received time of the last email that was synchronized to CRM for that mailbox. The FolderIds from which we want to get the email Ids are passed to Exchange Web Services (EWS).
3. GetItemsStep:  
This step gets the properties of the email that need to be synchronized by providing the email's Id and then creates the email in CRM. Each call to EWS gets a batch of such email.
4. GetAttachmentStep:  
For email that has attachments, this step gets the attachments from Exchange and then creates the email with the associated attachments in CRM.
5. UpdateExchangeItemsStep:  
Once the Exchange emails are added in CRM, this step updates the relevant CRM information (LinkStateTrackerState, LinkState, CrmId, RegardingObjectId, RegardingObjectName, RegardingObjectTypeCode) back in Exchange.
6. FindFolderStep:  
This step tries to locate folders, such as the Undeliverable folder, in Exchange. All Exchange email that could not be promoted in CRM is moved to the Undeliverable folder.
7. CreateFolderStep:  
If CRM cannot find an Undeliverable folder in Exchange, the folder is created.
8. MoveItemsStep:  
This step moves the email that could not be promoted in CRM to the Undeliverable Folder in Exchange. This email does not include the email that wasn't promoted due to the user's email handling settings, such as when **Track** is not set to **All email messages**.
9. DeleteItemsStep:

---

If the user has set their preferences to delete Exchange email, this step deletes processed email in CRM from the client that performs the synchronization, such as CRM App for Outlook or Microsoft Dynamics CRM for Outlook.

B. Manual tracking of email:

Besides the automated tracking, with the CRM for Outlook add-in installed the user also has the option to manually track an email from Outlook. When a user tracks the email in such a way the email gets marked for synchronization from one of the clients that supports synchronization, such as CRM App for Outlook or Microsoft Dynamics CRM for Outlook. These clients update the Exchange property of the item which is described in the [Outlook Synchronization in CRM 2015](#) white paper. To synchronize such email, two additional steps are performed before proceeding with all the steps mentioned earlier.

1. FindSearchFolderStep:

This step contacts Exchange to find if there is already a search folder created for filtering the email that needs to be synchronized to CRM. A search folder represents a persistent "always-on" search in a user's mailbox. This search folder uses the properties on the email which are sent when the user clicks Track. For more information about search folders, see [How to: Work with search folders by using EWS in Exchange](#).

2. CreateSearchFolderStep:

If such a search folder does not exist this step creates the relevant search folder.

Once the search folder is ready, emails are fetched and processed using a mechanism similar to the CRM automated tracking procedure.

### Tracking and untracking behaviors:

Email that arrive in the Inbox:

1. Will be synchronized to CRM.
2. The "Track" button may still show as "Track" even after synchronization.

Email that is synchronized as part of folder level tracking or by manually clicking "Track:"

1. Will be synchronized to CRM.
2. The "Track" button will change to "Untrack." For more information about this process, see the [Outlook Synchronization in CRM 2015](#) white paper.

Clicking "Untrack" manually:

1. Will remove the link between the Exchange email and the CRM email.
2. But both the emails will continue to exist, no email will be deleted.

Clicking "Track" after having tracked and untracked the email:

1. Since the email was previously tracked, both Exchange and CRM emails are present.
2. Clicking "Track" again will simply relink them.

### ***Queuing mechanisms, steps, calls to EWS, data merge, threading, and I/O ports***

Since all the Appointment, Contact, Task, and Email entities use the same Asynchronous Processing Service infrastructure, they share the same queuing mechanism that was described in the previous [Basic Overview](#) section.

This section describes the number of EWS calls that are made at every step of the synchronization process. Every EWS Async Begin request includes a callback method for EWS to call back. This callback method in turn calls the End method for that step.

### Outgoing email

| Step name            | EWS methods called  | EWS method call count per sync cycle for X number of emails | Callbacks to Async server from EWS for X number of emails |
|----------------------|---|---|---|
| CreateItemStep       | System.IAsyncResult BeginCreateItem()<br>CreateItemResponseType EndCreateItem()                         | 2 * X   | 1 * X   |
| AddDelegateStep      | System.IAsyncResult BeginAddDelegate()<br>AddDelegateResponseMessageType<br>EndAddDelegate()            | 2 * X   | 1 * X   |
| CreateAttachmentStep | System.IAsyncResult<br>BeginCreateAttachment()<br>CreateAttachmentResponseType<br>EndCreateAttachment() | 2 * X   | 1 * X   |
| SendItemStep         | System.IAsyncResult BeginSendItem()<br>SendItemResponseType EndSendItem()                               | 2 * X   | 1 * X   |
|                      |   | <b>Total EWS calls per sync cycle: 8 * X</b>                | <b>Total EWS calls per sync cycle: 4 * X</b>              |

### Incoming email

| Step name                   | EWS methods called   | EWS method call count per sync cycle for Y number of emails | Callbacks to Async server from EWS for Y number of emails |
|-----------------------------|--|---|---|
| GetInboxFolderHierarchyStep | System.IAsyncResult<br>BeginFindFolder()<br>FindFolderResponseType<br>EndFindFolder()          | 2 * Y   | 1 * Y   |
| FindItemsStep               | System.IAsyncResult BeginFindItem()<br>FindItemResponseType<br>EndFindItem()                   | 2 * Y   | 1 * Y   |
| GetItemsStep                | System.IAsyncResult BeginGetItem()<br>GetItemResponseType EndGetItem()                         | 2 * Y   | 1 * Y   |
| GetAttachmentStep           | System.IAsyncResult<br>BeginGetAttachment()<br>GetAttachmentResponseType<br>EndGetAttachment() | 2 * Y   | 1 * Y   |
| UpdateExchangeItemsStep     | System.IAsyncResult<br>BeginUpdateItem()<br>UpdateItemResponseType                             | 2 * Y   | 1 * Y   |

|                        |   |   |   |
|------------------------|---|---|---|
|                        | EndUpdateItem()   |   |   |
| FindFolderStep         | System.IAsyncResult<br>BeginFindFolder()<br>FindFolderResponseType<br>EndFindFolder()       | 2 * Y   | 1 * Y   |
| CreateFolderStep       | System.IAsyncResult<br>BeginCreateFolder()<br>CreateFolderResponseType<br>EndCreateFolder   | 2 * Y   | 1 * Y   |
| MoveItemsStep          | System.IAsyncResult<br>BeginMoveItem()<br>MoveItemResponseType<br>EndMoveItem()             | 2 * Y   | 1 * Y   |
| DeleteItemsStep        | System.IAsyncResult<br>BeginDeleteItem()<br>DeleteItemResponseType<br>EndDeleteItem()       | 2 * Y   | 1 * Y   |
| FindSearchFolderStep   | System.IAsyncResult<br>BeginFindFolder()<br>FindFolderResponseType<br>EndFindFolder()       | 2 * Y   | 1 * Y   |
| CreateSearchFolderStep | System.IAsyncResult<br>BeginCreateFolder()<br>CreateFolderResponseType<br>EndCreateFolder() | 2 * Y   | 1 * Y   |
|                        |   | <b>Total EWS calls per sync cycle: 22 * Y</b> | <b>Total EWS calls per sync cycle: 11 * Y</b> |

### Synchronization network loads

Therefore, if an organization has M mailboxes, and each mailbox has X outgoing emails and Y incoming emails, then one synchronization cycle for all the mailboxes would incur the follow network loads.

- Network load on EWS:  $M * ((8 * X) + (22 * Y))$
- Callback network load on Asynchronous Processing Service Server:  $M * ((4 * X) + (11 * Y))$

Each mailbox is processed every 5 minutes or 12 times an hour. So, the hourly network loads would be 12 times the above values.

## Tracking Appointments, Contacts, and Tasks

After mailboxes are processed from the mailbox queue, they are pushed to the server-side sync pipeline, which contains essentially 15 steps. Aside from the last step “commit”, the following 11 steps (including the twice

---

write to CRM step and update in Exchange step for data merging purpose) will be executed to process the data between Exchange and CRM.

## **Steps required for queuing mechanisms, calls to EWS, data merge, threading, and I/O ports**

Appointment, contact, tasks, and email share the same mailbox queue as incoming emails as described in the [Basic Overview](#) section.

1. **Synchronize folders (SyncFolderHierarchyStep):** An EWS call is made to search through the folder hierarchy in Exchange for each mailbox to get the Exchange folder Ids.
2. **Rebuild mapping for items (RebuildMappingStep):** As mentioned before, this step only happens when you first use server-side sync with existing data in Exchange.
3. **Sync folder item:** Makes an EWS call to grab all the items since the last time a synchronization process was performed on the folder, for each folder, that is received from the earlier **synchronize folders** step.
4. **Sync folder item properties:** Server-side sync will grab all of the folder item properties even if only one property has changed.
5. **AddRecurringAppointmentModifiedExceptionChangesStep:** Makes an EWS call to grab all the exceptional appointment from recurring appointment status.
6. **Fetch data:** Synchronization pipeline needs to sort through the actions (create/update/delete) for the items coming from Exchange and put them in an Exchange item collection. Next, CRM calls Prepare Sync, which is a service call to fetch back all the changes users made in the CRM web client. Then it sorts through actions (create/update/delete) for items coming from CRM, and puts them in a CRM item collection.
7. **Sync data:** Prepares data that needs to be synchronized based on what was collected during the **fetch data** step. The Exchange synchronization pipeline creates action queues for CRM-bound and Exchange-bound items.
8. **Write to CRM:** This process can process all items in actions queues, which we created in the last step, for the corresponding (create/update/delete/untrack/untrack and delete) actions we specified in the queue. While processing, we first iterate through all the properties to before we actually make the server call to proceed with the changes.
9. **Update in Exchange/Update in Exchange/Delete in Exchange:** Similar to writing in CRM, all the items are searched through for item property validation. Next, EWS finishes the update process from CRM to Exchange.
10. **UploadAttachmentsInExchangeStep:** If there's any attachments associated with appointments, this step executes the sending of the attachment with the appointment.
11. **Commit Step**

### **EWS calls**

| <b>Step name</b>        | <b>EWS methods called</b>                          | <b>EWS method call count per sync cycle</b> |
|-------------------------|--|---|
| SyncFolderHierarchyStep | BeginSyncFolderHierarchy<br>EndSyncFolderHierarchy | 3 (Contact/Calendar/Task)                   |

|  |  |                                   |
|--|--|-----------------------------------|
| Rebuild Mapping For items                            | FindItems                                  | 3 (Contact/Calendar/Task)         |
| Sync Folder Item                                     | BeginSyncFolderItems<br>EndSyncFolderItems | 3 (Contact/Calendar/Task)         |
| Sync Folder Item Properties                          | LoadPropertiesForItems                     | 3 (Contact/Calendar/Task)         |
| AddRecurringAppointmentModifiedExceptionChangesStep: | FindAppointments                           | 1                                 |
| Fetch Data   | N/A  |                                   |
| Sync Data  | N/A  |                                   |
| Write To Crm   | N/A  |                                   |
| Create in Exchange                                   | CreateItems                                | Up to<br>3(Contact/Calendar/Task) |
| Update in Exchange                                   | UpdateItems                                | Up to 3 (Contact/Calendar/Task)   |
| Delete in Exchange                                   | DeleteItems                                | Up to 3 (Contact/Calendar/Task)   |
| UploadAttachmentsInExchangeStep                      | AddFileAttachment                          | 1 X N attachments                 |
| <b>Total EWS calls</b>                               |  | <b>16 + N (attachments)</b>       |

### ***Sync filters and server-side sync impact***

Synchronization filters lets users choose which appointments, contacts, and tasks to synchronize between CRM and Exchange. Email is not included in the synchronization filters because email synchronization is controlled when the email is created in CRM, whether the user is on the recipient list or not.

## Supportability

### ***Errors fields in the MailboxBase table***

The MailboxBase table in the organization database stores the information for each mailbox including the error information for the last ACT and incoming email processing synchronization cycle. This is available in both Dynamics CRM on-premises and Dynamics CRM Online deployments. The error information can help on-premises deployment administrators diagnose failures in mailboxes and mailbox items as well as synchronization state and statistics.

- **TransientFailureCount:** An integer which stores the count of transient errors for incoming email, outgoing email, and ACT processing. These counts are bit-shifted to fit into the same integer value. The value of the transient error count for all these three processing types goes from 0-15 (4 bits), with 0-2 considered as good health, 2-7 considered as average health and anything above 7 considered as poor health. The transient error count is a contiguous count in the sense that if any of the processing types has a successful sync cycle; this value is reset to 0 for that processing type.

- 
- **LastSuccessfulSyncCompletedOn:** This stores the timestamp for the completion of the last successful synchronization cycle for ACT processing.
  - **ProcessedTimes:** Stores the count of the number of times the mailbox was processed.
  - **AverageTotalDuration:** Stores the running average of the total time spent for the mailbox in ACT synchronization processing.
  - **LastSyncErrorOccuredOn:** Stores the timestamp of the error that occurred during last ACT processing cycle.
  - **ItemsFailedForLastSync:** Stores the count of ACT items whose processing failed during last synchronization.
  - **LastSyncErrorMachineName:** Stores the name of the server running the Asynchronous Processing Service that was processing the mailbox when the last ACT synchronization error occurred.
  - **LastSyncErrorCode:** Stores the CRM trace code for the last ACT synchronization error that happened.
  - **ItemsProcessedForLastSync:** Stores the count of the total number of ACT items processed during the last sync cycle (failed + succeeded).
  - **LastSyncStartedOn:** Stores the timestamp when processing started for the mailbox for the last synchronization cycle.
  - **LastSyncError:** Stores 2408 bytes of the last synchronization error stack trace for ACT processing.
  - **LastSyncErrorCount:** Stores the contiguous count of the last synchronization error that occurred during ACT processing.
  - **LastDuration:** Stores the duration of the last ACT processing synchronization cycle.
  - **ProcessingLastAttemptedOn:** Stores the time stamp of when the mailbox was picked up and put in the asynchronous processing queue for processing.
  - **ForcedUnlockCount:** A counter used to determine how many times the mailbox was forcibly unlocked. Mailboxes are forcibly unlocked periodically when the organization setting `IsMailboxForcedUnlockingEnabled` is set to true.
  - **LastMailboxForcedUnlockOccuredOn:** This column stores the time stamp of when the last time the Mailbox was forced to unlock.
  - **VerboseLoggingEnabled:** This bit is set when the system administrator sets a mailbox for verbose logging using the Ribbon button on the Mailbox Edit page. When this bit is set, the mailbox starts logging verbose information.

These MailboxBase table columns can help support teams identify possible issues with mailboxes. All the columns prefixed with the 'LastSync' keyword have the information about the last synchronization cycle. These columns also store the error traces, if any, for the last synchronization cycle. The information can be used by operations teams and support professionals to identify whether a mailbox is working as expected or is failing.

### ***Performance table***

The MailboxStatistics table stores performance-related information for each ACT, incoming email, and outgoing email processing cycle. Because of the high volume of data in the table, the data is periodically purged. Purging is time dependent on the deployment properties settings.

- **ItemsFailed:** Count of the number of items that failed in the last synchronization cycle.
- **MailboxProcessStartedOn:** Time stamp of when mailbox processing started.
- **CrmlItemsBacklog:** The number of CRM items still left for processing after the synchronization cycle.
- **MailboxId:** Id of the mailbox processed during this synchronization cycle.

- 
- **MachineName:** Name of the server running the Asynchronous Processing Service where the synchronization cycle was executed.
  - **MailboxProcessCompletedOn:** Time stamp of when mailbox processing finished for the synchronization cycle.
  - **OperationTypeId:** Type of mailbox operation. Incoming email is 0, outgoing email is 1, and ACT is 2.
  - **ProcessResult:** Stores information when mailbox processing failed. Transient and permanent errors are 0, success is 1.
  - **ItemsProcessed:** Stores the count of the total items processed during the last synchronization cycle.
  - **MailboxProcessScheduledOn:** Stores the time stamp of when the mailbox was supposed to be processed (ReceivingPostponedUntil/ReceivingPostponedUntilForACT).
  - **IndividualStepDurations:** Stores the time it took for the individual step execution during ACT processing.
  - **ScheduledTimeIntervallnMinutes:** Stores the time in minutes when the mailbox was scheduled for processing (schedule time - processing start time).
  - **ProcessTimeIntervallnMinutes:** Stores the duration in minutes of mailbox processing (processing start time - processing completed time).

## **Tracing**

Server-side sync uses the following tracing mechanisms:

1. **Logfile:** Server-side sync platform-level trace information is recorded to the Asynchronous Processing Service log files. Depending on which server running the Asynchronous Processing Service the error occurred and the approximate time of when the error occurred, server-side sync can trace back the entire error stack from the log files. By default, log files are stored in the \Microsoft Dynamics CRM\Trace folder. More information: [Microsoft Dynamics CRM server-level tracing](#)
2. **Alert Wall (TraceLogBase table):** The TraceLogBase table along with its helper tables TraceAssociationBase and TraceRegardingBase, store the information that is relayed out to user on the Alert wall. The TraceLogBase table can also be very helpful in debugging processing failures because it stores per mailbox information for all errors that occur, on which server they occurred, and the time of the error. Using these three pieces of information can identify the correct log file to view and find the entire call stack for the error.
3. **Telemetry:** In addition to storing the error information in the organization database, server-side sync also sends error information, such as ErrorSource, ErrorScope, TraceSnippet, ServerErrorCode, CrmTraceCode, MachineName, and Timestamp. These pieces of information can help you find the preliminary data about errors. The information provided also points to the trace files that store the full error details.

## **TraceLogBase table**

As mentioned earlier, the TraceLogBase table is used to store the error Information for server-side sync that is surfaced on the alert wall. The TraceLogBase table can help identify the error code encountered that can then be mapped to the static mapping of the error code in CRM. Also, this table stores the name of the computer and the time when the error occurred and can be useful in identifying the exact traces for the error.

A T-SQL query can display the errors that have occurred in a deployment and, if needed, retrieve the correct traces associated with the error.

---

```
select TraceCode,RegardingObjectIdName,RegardingObjectId, MachineName, ModifiedOn from TraceLog
order by ModifiedOn desc
```

## ***ETW Tracing***

[Event Tracing for Windows \(ETW\)](#) is used to store predominantly performance counters and is also used by the system to send data for telemetry.

## **Performance Counters**

- AddRecurringAppointmentModifiedExceptionChangesStepAverageLatency = "Server-side synchronization: Add Recurring Appointment Modified Exception Changes Average Latency"
- AddRecurringAppointmentModifiedExceptionChangesStepAverageLatencyBase = "Server-side synchronization: Add Recurring Appointment Modified Exception Changes Average Latency Base"
- AddRecurringAppointmentModifiedExceptionChangesStepAverageLatencyBaseHelp = "Add recurring appointment modified exception changes average latency base"
- AddRecurringAppointmentModifiedExceptionChangesStepAverageLatencyHelp = "Add recurring appointment modified exception changes average latency"
- CommitStepAverageLatency = "Server-side synchronization: Commit Average Latency"
- CommitStepAverageLatencyBase = "Server-side synchronization: Commit Average Latency Base"
- CommitStepAverageLatencyBaseHelp = "Commit average latency base"
- CommitStepAverageLatencyHelp = "Commit average latency"
- CreateInExchangeStepAverageLatency = "Server-side synchronization: Create in Exchange Average Latency"
- CreateInExchangeStepAverageLatencyBase = "Server-side synchronization: Create in Exchange Average Latency Base"
- CreateInExchangeStepAverageLatencyBaseHelp = "Create in Exchange average latency base"
- CreateInExchangeStepAverageLatencyHelp = "Create in Exchange average latency"
- DeleteInExchangeStepAverageLatency = "Server-side synchronization: Delete in Exchange Average Latency"
- DeleteInExchangeStepAverageLatencyBase = "Server-side synchronization: Delete in Exchange Average Latency Base"
- DeleteInExchangeStepAverageLatencyBaseHelp = "Delete in Exchange average latency base"
- DeleteInExchangeStepAverageLatencyHelp = "Delete in Exchange average latency"
- DupDetectionThroughputHelp = "Number of records processed for duplicate detection per second";
- DupDetectionThroughputName = "Bulk Detect Duplicate Rate (Throughput)"
- EmailConnectorExchangeAutoDiscoverAverageDuration = "Server-side synchronization (Exchange Auto-discover): Average time taken for auto-discovery"
- EmailConnectorExchangeAutoDiscoverAverageDurationBase = "Server-side synchronization (Exchange Auto-discover): Average time taken for auto-discovery base"
- EmailConnectorExchangeAutoDiscoverAverageDurationBaseHelp = "Average time taken for auto-discovery of EWS endpoint (base)"

- 
- EmailConnectorExchangeAutoDiscoverAverageDurationHelp = "Average time taken for auto-discovery of EWS endpoint"
  - EmailConnectorExchangeAutoDiscoverCacheHit = "Server-side synchronization (Exchange Auto-discover): Cache Hit"
  - EmailConnectorExchangeAutoDiscoverCacheHitHelp = "Total number of auto discover requests that were fulfilled using the cache"
  - EmailConnectorExchangeAutoDiscoverCacheMiss = "Server-side synchronization (Exchange Auto-discover): Cache Miss"
  - EmailConnectorExchangeAutoDiscoverCacheMissHelp = "Total number of auto discover requests that could not be fulfilled using the cache"
  - EmailConnectorExchangeAutoDiscoverConnections = "Server-side synchronization (Exchange Auto-discover): Current number of connections"
  - EmailConnectorExchangeAutoDiscoverConnectionsHelp = "Current number of connections being used for auto-discovery of EWS endpoint"
  - EmailConnectorExchangeAutoDiscoverFailure = "Server-side synchronization (Exchange auto-discover): Discoveries Failed"
  - EmailConnectorExchangeAutoDiscoverFailureHelp = "Total number of auto discover calls that failed";
  - EmailConnectorExchangeAutoDiscoverSuccess = "Server-side synchronization (Exchange auto-discover): Discoveries Successful";
  - EmailConnectorExchangeAutoDiscoverSuccessHelp = "Total number of auto discover calls that were successful"
  - EmailConnectorIncomingEmailsDelivered = "Server-side synchronization (incoming): Emails delivered"
  - EmailConnectorIncomingEmailsDeliveredHelp = "Total number of incoming emails delivered"
  - EmailConnectorIncomingEmailsDeliveredThroughput = "Server-side synchronization (incoming): Items delivered Throughput"
  - EmailConnectorIncomingEmailsDeliveredThroughputHelp = "Throughput of delivered incoming mail items"
  - EmailConnectorIncomingEmailsDiscarded = "Server-side synchronization (Incoming): Items Discarded"
  - EmailConnectorIncomingEmailsDiscardedHelp = "Total number of incoming mail items discarded"
  - EmailConnectorIncomingEmailsDiscardedThroughput = "Server-side synchronization (Incoming): Items Discarded Throughput"
  - EmailConnectorIncomingEmailsDiscardedThroughputHelp = "Throughput of discarded incoming mail items"
  - EmailConnectorIncomingEmailsEwsAverageConnectionDuration = "Server-side synchronization (Incoming): Average duration of an EWS connection"
  - EmailConnectorIncomingEmailsEwsAverageConnectionDurationBase = "Server-side synchronization (Incoming): Average duration of an EWS connection base"
  - EmailConnectorIncomingEmailsEwsAverageConnectionDurationBaseHelp = "Average duration of an EWS connection for incoming email (base)"
  - EmailConnectorIncomingEmailsEwsAverageConnectionDurationHelp = "Average duration of an EWS connection for incoming email"

- 
- EmailConnectorIncomingEmailsEwsBytesReceivedThroughput = "Server-side synchronization (Incoming): Throughput of bytes received in EWS responses"
  - EmailConnectorIncomingEmailsEwsBytesReceivedThroughputHelp = "Throughput of bytes received in EWS responses while processing incoming mail items"
  - EmailConnectorIncomingEmailsEwsBytesSentThroughput = "Server-side synchronization (Incoming): Throughput of bytes sent in EWS requests"
  - EmailConnectorIncomingEmailsEwsBytesSentThroughputHelp = "Throughput of bytes sent in EWS requests while processing incoming mail items"
  - EmailConnectorIncomingEmailsEwsConnections = "Server-side synchronization (Incoming): Current number of EWS Connections"
  - EmailConnectorIncomingEmailsEwsConnectionsHelp = "Current number of EWS connections being used for processing incoming mail items"
  - EmailConnectorIncomingEmailsFailed = "Server-side synchronization (Incoming): Items Failed"
  - EmailConnectorIncomingEmailsFailedHelp = "Total number of incoming mail items failed";
  - EmailConnectorIncomingEmailsFailedThroughput = "Server-side synchronization (Incoming): Items Failed Throughput"
  - EmailConnectorIncomingEmailsFailedThroughputHelp = "Throughput of failed incoming mail items";
  - EmailConnectorIncomingEmailsPop3AverageConnectionDuration = "Server-side synchronization (Incoming): Average duration of a POP3 connection"
  - EmailConnectorIncomingEmailsPop3AverageConnectionDurationBase = "Server-side synchronization (Incoming): Average duration of a POP3 connection base"
  - EmailConnectorIncomingEmailsPop3AverageConnectionDurationBaseHelp = "Average duration of a POP3 connection (base)"
  - EmailConnectorIncomingEmailsPop3AverageConnectionDurationHelp = "Average duration of a POP3 connection"
  - EmailConnectorIncomingEmailsPop3BytesReceivedThroughput = "Server-side synchronization (Incoming): Throughput of bytes received via POP3 protocol"
  - EmailConnectorIncomingEmailsPop3BytesReceivedThroughputHelp = "Throughput of bytes received via POP3 protocol while processing incoming mail items"
  - EmailConnectorIncomingEmailsPop3BytesSentThroughput = "Server-side synchronization (Incoming): Throughput of bytes sent via POP3 protocol"
  - EmailConnectorIncomingEmailsPop3BytesSentThroughputHelp = "Throughput of bytes sent via POP3 protocol while processing incoming mail items"
  - EmailConnectorIncomingEmailsPop3Connections = "Server-side synchronization (Incoming): Current number of POP3 Connections"
  - EmailConnectorIncomingEmailsPop3ConnectionsHelp = "Current number of POP3 connections being used for processing incoming mail items"
  - EmailConnectorIncomingEmailsProcessed = "Server-side synchronization (Incoming): Emails Processed"
  - EmailConnectorIncomingEmailsProcessedHelp = "Total number of incoming emails processed"
  - EmailConnectorIncomingEmailsProcessedThroughput = "Server-side synchronization (Incoming): Items Processed Throughput"

- 
- EmailConnectorIncomingEmailsProcessedThroughputHelp = "Throughput of processed incoming mail items"
  - EmailConnectorIncomingEmailsSuspected = "Server-side synchronization (Incoming): Items Suspected"
  - EmailConnectorIncomingEmailsSuspectedHelp = "Total number of incoming mail items potentially corrupted"
  - EmailConnectorIncomingEmailsSuspectedThroughput = "Server-side synchronization (Incoming): Items Suspected Throughput"
  - EmailConnectorIncomingEmailsSuspectedThroughputHelp = "Throughput of potentially corrupted incoming mail items"
  - EmailConnectorIncomingEmailsTransientErrors = "Server-side synchronization (Incoming): Transient Errors"
  - EmailConnectorIncomingEmailsTransientErrorsHelp = "Total number of transient errors in incoming mail items"
  - EmailConnectorIncomingEmailsTransientErrorsThroughput = "Server-side synchronization (Incoming): Transient Errors Throughput"
  - EmailConnectorIncomingEmailsTransientErrorsThroughputHelp = "Throughput of transient errors in incoming mail items"
  - EmailConnectorOutgoingEmailsDelivered = "Server-side synchronization (Outgoing): Items Delivered"
  - EmailConnectorOutgoingEmailsDeliveredHelp = "Total number of outgoing mail items delivered"
  - EmailConnectorOutgoingEmailsDeliveredThroughput = "Server-side synchronization (Outgoing): Items Delivered Throughput"
  - EmailConnectorOutgoingEmailsDeliveredThroughputHelp = "Throughput of delivered outgoing mail items"
  - EmailConnectorOutgoingEmailsEwsAverageConnectionDuration = "Server-side synchronization (Outgoing): Average duration of an EWS connection"
  - EmailConnectorOutgoingEmailsEwsAverageConnectionDurationBase = "Server-side synchronization (Outgoing): Average duration of an EWS connection base"
  - EmailConnectorOutgoingEmailsEwsAverageConnectionDurationBaseHelp = "Average duration of an EWS connection for outgoing email (base)"
  - EmailConnectorOutgoingEmailsEwsAverageConnectionDurationHelp = "Average duration of an EWS connection for outgoing email"
  - EmailConnectorOutgoingEmailsEwsBytesReceivedThroughput = "Server-side synchronization (Outgoing): Throughput of bytes received in EWS responses"
  - EmailConnectorOutgoingEmailsEwsBytesReceivedThroughputHelp = "Throughput of bytes received in EWS responses while sending outgoing email"
  - EmailConnectorOutgoingEmailsEwsBytesSentThroughput = "Server-side synchronization (Outgoing): Throughput of bytes sent in EWS requests"
  - EmailConnectorOutgoingEmailsEwsBytesSentThroughputHelp = "Throughput of bytes sent in EWS requests while sending outgoing email"
  - EmailConnectorOutgoingEmailsEwsConnections = "Server-side synchronization (Outgoing): Current number of EWS Connections"

- 
- EmailConnectorOutgoingEmailsEwsConnectionsHelp = "Current number of EWS connections being used for sending outgoing email"
  - EmailConnectorOutgoingEmailsFailed = "Server-side synchronization (Outgoing): Items Failed"
  - EmailConnectorOutgoingEmailsFailedHelp = "Total number of outgoing mail items failed"
  - EmailConnectorOutgoingEmailsFailedThroughput = "Server-side synchronization (Outgoing): Items Failed Throughput"
  - EmailConnectorOutgoingEmailsFailedThroughputHelp = "Throughput of failed outgoing mail items"
  - EmailConnectorOutgoingEmailsProcessed = "Server-side synchronization (Outgoing): Items Processed"
  - EmailConnectorOutgoingEmailsProcessedHelp = "Total number of outgoing mail items processed"
  - EmailConnectorOutgoingEmailsProcessedThroughput = "Server-side synchronization (Outgoing): Items Processed Throughput"
  - EmailConnectorOutgoingEmailsProcessedThroughputHelp = "Throughput of processed outgoing mail items"
  - EmailConnectorOutgoingEmailsSmtpAverageConnectionDuration = "Server-side synchronization (Outgoing): Average duration of an SMTP connection"
  - EmailConnectorOutgoingEmailsSmtpAverageConnectionDurationBase = "Server-side synchronization (Outgoing): Average duration of an SMTP connection base"
  - EmailConnectorOutgoingEmailsSmtpAverageConnectionDurationBaseHelp = "Average duration of an SMTP connection (base)"
  - EmailConnectorOutgoingEmailsSmtpAverageConnectionDurationHelp = "Average duration of an SMTP connection"
  - EmailConnectorOutgoingEmailsSmtpBytesSentThroughput = "Server-side synchronization (Outgoing): Throughput of bytes sent via SMTP protocol"
  - EmailConnectorOutgoingEmailsSmtpBytesSentThroughputHelp = "Approximate throughput of bytes sent via SMTP protocol while sending outgoing email"
  - EmailConnectorOutgoingEmailsSmtpConnections = "Server-side synchronization (Outgoing): Current number of SMTP Connections"
  - EmailConnectorOutgoingEmailsSmtpConnectionsHelp = "Current number of SMTP connections being used for sending outgoing email"
  - EmailConnectorOutgoingEmailsTransientErrors = "Server-side synchronization (Outgoing): Transient Errors"
  - EmailConnectorOutgoingEmailsTransientErrorsHelp = "Total number of transient errors in outgoing mail items";
  - EmailConnectorOutgoingEmailsTransientErrorsThroughput = "Server-side synchronization (Outgoing): Transient Errors Throughput"
  - EmailConnectorOutgoingEmailsTransientErrorsThroughputHelp = "Throughput of transient errors in outgoing mail items"
  - EmailConnectorSynchronousDelays = "Server-side synchronization: Synchronous Delays"
  - EmailConnectorSynchronousDelaysHelp = "Total number of times a synchronous delay was added while email processing"
  - ExchangeSynchronizationACTDuration = "Server-side synchronization: ACT Average Processing Duration"

- 
- ExchangeSynchronizationACTDurationBase = "Server-side synchronization: ACT Average Processing Duration Base"
  - ExchangeSynchronizationACTDurationBaseHelp = "Average duration of ACT processing (base)"
  - ExchangeSynchronizationACTDurationHelp = "Per-mailbox duration for ACT processing"
  - ExchangeSynchronizationACTEwsAverageConnectionDuration = "Server-side synchronization: Average Duration of an EWS Connection for ACTs"
  - ExchangeSynchronizationACTEwsAverageConnectionDurationBase = "Server-side synchronization: Average Duration of an EWS Connection Base for ACTs"
  - ExchangeSynchronizationACTEwsAverageConnectionDurationBaseHelp = "Average duration of an EWS connection for appointments, contacts, and tasks (base)"
  - ExchangeSynchronizationACTEwsAverageConnectionDurationHelp = "Average duration of an EWS connection for appointments, contacts, and tasks"
  - ExchangeSynchronizationACTEwsBytesReceivedThroughput = "Server-side synchronization: Throughput of Bytes Received in EWS Requests for ACTs"
  - ExchangeSynchronizationACTEwsBytesReceivedThroughputHelp = "Throughput of bytes received in EWS requests while processing appointments, contacts, and tasks"
  - ExchangeSynchronizationACTEwsBytesSentThroughput = "Server-side synchronization: Throughput of Bytes Sent in EWS Requests for ACTs"
  - ExchangeSynchronizationACTEwsBytesSentThroughputHelp = "Throughput of bytes sent in EWS requests while processing appointments, contacts, and tasks"
  - ExchangeSynchronizationACTEwsConnections = "Server-side synchronization: Current Number of EWS Connections for ACTs"
  - ExchangeSynchronizationACTEwsConnectionsHelp = "Current number of EWS connections being used for processing appointments, contacts, and tasks"
  - ExchangeSynchronizationACTFailed = "Server-side synchronization: ACTs Failed"
  - ExchangeSynchronizationACTFailedHelp = "Total number of appointments, contacts, and tasks that failed"
  - ExchangeSynchronizationACTFailedThroughput = "Server-side synchronization: ACTs Failed Throughput"
  - ExchangeSynchronizationACTFailedThroughputHelp = "Throughput of appointments, contacts, and tasks that failed"
  - ExchangeSynchronizationACTLastDuration = "Server-side synchronization: ACT Processing Last Duration"
  - ExchangeSynchronizationACTLastDurationHelp = "Average duration of ACT processing"
  - ExchangeSynchronizationACTMailboxesProcessed = "Server-side synchronization: Mailboxes processed for ACT"
  - ExchangeSynchronizationACTMailboxesProcessedHelp = "Total number of mailboxes processed for appointments, contacts, and tasks"
  - ExchangeSynchronizationACTProcessed = "Server-side synchronization: ACTs Processed"
  - ExchangeSynchronizationACTProcessedHelp = "Total number of appointments, contacts, and tasks processed"
  - ExchangeSynchronizationACTProcessedThroughput = "Server-side synchronization: ACTs Processed Throughput"

- 
- ExchangeSynchronizationACTProcessedThroughputHelp = "Throughput of processed appointments, contacts, and tasks"
  - ExchangeSynchronizationACTSynchronized = "Server-side synchronization: ACTs Synchronized"
  - ExchangeSynchronizationACTSynchronizedHelp = "Total number of appointments, contacts, and tasks synchronized"
  - ExchangeSynchronizationACTSynchronizedThroughput = "Server-side synchronization: ACTs Synchronized Throughput"
  - ExchangeSynchronizationACTSynchronizedThroughputHelp = "Throughput of appointments, contacts, and tasks synchronized"
  - ExchangeSynchronizationACTTransientErrors = "Server-side synchronization: Transient Errors for ACTs"
  - ExchangeSynchronizationACTTransientErrorsHelp = "Total number of transient errors for appointments, contacts, and tasks"
  - ExchangeSynchronizationACTTransientErrorsThroughput = "Server-side synchronization: ACTs Transient Errors Throughput"
  - ExchangeSynchronizationACTTransientErrorsThroughputHelp = "Throughput of transient errors for appointments, contacts, and tasks"
  - ExchangeSynchronizationAppointmentsEwsAverageConnectionDuration = "Server-side synchronization: Average Duration of an EWS Connection for Appointments"
  - ExchangeSynchronizationAppointmentsEwsAverageConnectionDurationBase = "Server-side synchronization: Average duration of an EWS connection base for Appointments"
  - ExchangeSynchronizationAppointmentsEwsAverageConnectionDurationBaseHelp = "Average duration of an EWS connection for appointments (base)"
  - ExchangeSynchronizationAppointmentsEwsAverageConnectionDurationHelp = "Average duration of an EWS connection for appointments"
  - ExchangeSynchronizationAppointmentsEwsBytesReceivedThroughput = "Server-side synchronization: Throughput of Bytes Received in EWS Requests for Appointments"
  - ExchangeSynchronizationAppointmentsEwsBytesReceivedThroughputHelp = "Throughput of bytes received in EWS requests while processing appointments"
  - ExchangeSynchronizationAppointmentsEwsBytesSentThroughput = "Server-side synchronization: Throughput of bytes sent in EWS requests for Appointments"
  - ExchangeSynchronizationAppointmentsEwsBytesSentThroughputHelp = "Throughput of bytes sent in EWS requests while processing appointments"
  - ExchangeSynchronizationAppointmentsEwsConnections = "Server-side synchronization: Current Number of EWS Connections for Appointments"
  - ExchangeSynchronizationAppointmentsEwsConnectionsHelp = "Current number of EWS connections being used for processing appointments"
  - ExchangeSynchronizationAppointmentsFailed = "Server-side synchronization: Appointments Failed"
  - ExchangeSynchronizationAppointmentsFailedHelp = "Total number of appointments that failed"
  - ExchangeSynchronizationAppointmentsFailedThroughput = "Server-side synchronization: Appointments Failed Throughput"
  - ExchangeSynchronizationAppointmentsFailedThroughputHelp = "Throughput of appointments that failed"

- 
- ExchangeSynchronizationAppointmentsProcessed = "Server-side synchronization: Appointments Processed"
  - ExchangeSynchronizationAppointmentsProcessedHelp = "Total number of appointments processed"
  - ExchangeSynchronizationAppointmentsProcessedThroughput = "Server-side synchronization: Appointments Processed Throughput"
  - ExchangeSynchronizationAppointmentsProcessedThroughputHelp = "Throughput of processed appointments"
  - ExchangeSynchronizationAppointmentsSynchronized = "Server-side synchronization: Appointments Synchronized"
  - ExchangeSynchronizationAppointmentsSynchronizedHelp = "Total number of appointments synchronized"
  - ExchangeSynchronizationAppointmentsSynchronizedThroughput = "Server-side synchronization: Appointments Synchronized Throughput"
  - ExchangeSynchronizationAppointmentsSynchronizedThroughputHelp = "Throughput of appointments synchronized"
  - ExchangeSynchronizationAppointmentsTransientErrors = "Server-side synchronization: Transient Errors for Appointments"
  - ExchangeSynchronizationAppointmentsTransientErrorsHelp = "Total number of transient errors for appointments"
  - ExchangeSynchronizationAppointmentsTransientErrorsThroughput = "Server-side synchronization: Appointments Transient Errors Throughput"
  - ExchangeSynchronizationAppointmentsTransientErrorsThroughputHelp = "Throughput of transient errors for appointments"
  - ExchangeSynchronizationContactsEwsAverageConnectionDuration = "Server-side synchronization: Average Duration of an EWS Connection for Contacts"
  - ExchangeSynchronizationContactsEwsAverageConnectionDurationBase = "Server-side synchronization: Average duration of an EWS connection base for Contacts"
  - ExchangeSynchronizationContactsEwsAverageConnectionDurationBaseHelp = "Average duration of an EWS connection for contacts (base)"
  - ExchangeSynchronizationContactsEwsAverageConnectionDurationHelp = "Average duration of an EWS connection for contacts"
  - ExchangeSynchronizationContactsEwsBytesReceivedThroughput = "Server-side synchronization: Throughput of Bytes Received in EWS Requests for Contacts"
  - ExchangeSynchronizationContactsEwsBytesReceivedThroughputHelp = "Throughput of bytes received in EWS requests while processing contacts"
  - ExchangeSynchronizationContactsEwsBytesSentThroughput = "Server-side synchronization: Throughput of Bytes Sent in EWS Requests for Contacts"
  - ExchangeSynchronizationContactsEwsBytesSentThroughputHelp = "Throughput of bytes sent in EWS requests while processing contacts"
  - ExchangeSynchronizationContactsEwsConnections = "Server-side synchronization: Current Number of EWS Connections for Contacts"
  - ExchangeSynchronizationContactsEwsConnectionsHelp = "Current number of EWS connections being used for processing contacts"

- 
- ExchangeSynchronizationContactsFailed = "Server-side synchronization: Contacts Failed"
  - ExchangeSynchronizationContactsFailedHelp = "Total number of contacts that failed"
  - ExchangeSynchronizationContactsFailedThroughput = "Server-side synchronization: Contacts Failed Throughput"
  - ExchangeSynchronizationContactsFailedThroughputHelp = "Throughput of contacts that failed"
  - ExchangeSynchronizationContactsProcessed = "Server-side synchronization: Contacts Processed"
  - ExchangeSynchronizationContactsProcessedHelp = "Total number of contacts processed"
  - ExchangeSynchronizationContactsProcessedThroughput = "Server-side synchronization: Contacts Processed Throughput"
  - ExchangeSynchronizationContactsProcessedThroughputHelp = "Throughput of processed contacts"
  - ExchangeSynchronizationContactsSynchronized = "Server-side synchronization: Contacts Synchronized"
  - ExchangeSynchronizationContactsSynchronizedHelp = "Total number of contacts synchronized"
  - ExchangeSynchronizationContactsSynchronizedThroughput = "Server-side synchronization: Contacts Synchronized Throughput"
  - ExchangeSynchronizationContactsSynchronizedThroughputHelp = "Throughput of contacts synchronized"
  - ExchangeSynchronizationContactsTransientErrors = "Server-side synchronization: Transient Errors for Contacts"
  - ExchangeSynchronizationContactsTransientErrorsHelp = "Total number of transient errors for contacts";
  - ExchangeSynchronizationContactsTransientErrorsThroughput = "Server-side synchronization: Contacts Transient Errors Throughput"
  - ExchangeSynchronizationContactsTransientErrorsThroughputHelp = "Throughput of transient errors for contacts"
  - ExchangeSynchronizationCrmItemsBacklog = "Server-side synchronization: Current Number of ACTs Have Not Synchronized From CRM Server"
  - ExchangeSynchronizationCrmItemsBacklogHelp = "Current number of contacts, appointments and tasks have not synchronized from CRM server"
  - ExchangeSynchronizationTasksEwsAverageConnectionDuration = "Server-side synchronization: Average Duration of an EWS Connection for Tasks"
  - ExchangeSynchronizationTasksEwsAverageConnectionDurationBase = "Server-side synchronization: Average duration of an EWS connection base for Tasks"
  - ExchangeSynchronizationTasksEwsAverageConnectionDurationBaseHelp = "Average duration of an EWS connection for tasks (base)"
  - ExchangeSynchronizationTasksEwsAverageConnectionDurationHelp = "Average duration of an EWS connection for tasks"
  - ExchangeSynchronizationTasksEwsBytesReceivedThroughput = "Server-side synchronization: Throughput of Bytes Received in EWS Requests for Tasks"
  - ExchangeSynchronizationTasksEwsBytesReceivedThroughputHelp = "Throughput of bytes received in EWS requests while processing tasks"
  - ExchangeSynchronizationTasksEwsBytesSentThroughput = "Server-side synchronization: Throughput of Bytes Sent in EWS Requests for Tasks"

- 
- ExchangeSynchronizationTasksEwsBytesSentThroughputHelp = "Throughput of bytes sent in EWS requests while processing tasks"
  - ExchangeSynchronizationTasksEwsConnections = "Server-side synchronization: Current Number of EWS Connections for Tasks"
  - ExchangeSynchronizationTasksEwsConnectionsHelp = "Current number of EWS connections being used for processing tasks"
  - ExchangeSynchronizationTasksFailed = "Server-side synchronization: Tasks Failed"
  - ExchangeSynchronizationTasksFailedHelp = "Total number of tasks that failed"
  - ExchangeSynchronizationTasksFailedThroughput = "Server-side synchronization: Tasks Failed Throughput"
  - ExchangeSynchronizationTasksFailedThroughputHelp = "Throughput of tasks that failed"
  - ExchangeSynchronizationTasksProcessed = "Server-side synchronization: Tasks Processed"
  - ExchangeSynchronizationTasksProcessedHelp = "Total number of tasks processed"
  - ExchangeSynchronizationTasksProcessedThroughput = "Server-side synchronization: Tasks Processed Throughput"
  - ExchangeSynchronizationTasksProcessedThroughputHelp = "Throughput of processed tasks"
  - ExchangeSynchronizationTasksSynchronized = "Server-side synchronization: Tasks Synchronized"
  - ExchangeSynchronizationTasksSynchronizedHelp = "Total number of tasks synchronized"
  - ExchangeSynchronizationTasksSynchronizedThroughput = "Server-side synchronization: Tasks Synchronized Throughput"
  - ExchangeSynchronizationTasksSynchronizedThroughputHelp = "Throughput of tasks synchronized"
  - ExchangeSynchronizationTasksTransientErrors = "Server-side synchronization: Transient Errors for Tasks"
  - ExchangeSynchronizationTasksTransientErrorsHelp = "Total number of transient errors for tasks"
  - ExchangeSynchronizationTasksTransientErrorsThroughput = "Server-side synchronization: Tasks Transient Errors Throughput"
  - ExchangeSynchronizationTasksTransientErrorsThroughputHelp = "Throughput of transient errors for tasks"
  - FetchDataStepAverageLatency = "Server-side synchronization: Fetch Data Average Latency"
  - FetchDataStepAverageLatencyBase = "Server-side synchronization: Fetch Data Average Latency Base"
  - FetchDataStepAverageLatencyBaseHelp = "Fetch data average latency base"
  - FetchDataStepAverageLatencyHelp = "Fetch data average latency";
  - LoadPropertiesForItemsStepAverageLatency = "Server-side synchronization: Load Properties For Items Average Latency"
  - LoadPropertiesForItemsStepAverageLatencyBase = "Server-side synchronization: Load Properties For Items Average Latency Base"
  - LoadPropertiesForItemsStepAverageLatencyBaseHelp = "Load properties for items average latency base"
  - LoadPropertiesForItemsStepAverageLatencyHelp = "Load properties for items average latency"
  - RebuildMappingStepAverageLatency = "Server-side synchronization: Rebuild Mapping Step Average Latency"

- 
- RebuildMappingStepAverageLatencyBase = "Server-side synchronization: Rebuild Mapping Step Average Latency Base"
  - RebuildMappingStepAverageLatencyBaseHelp = "Rebuild mapping step average latency base"
  - RebuildMappingStepAverageLatencyHelp = "Rebuild mapping step average latency"
  - SyncDataStepAverageLatency = "Server-side synchronization: Sync Data Average Latency"
  - SyncDataStepAverageLatencyBase = "Server-side synchronization: Sync Data Average Latency Base"
  - SyncDataStepAverageLatencyBaseHelp = "Sync data average latency base"
  - SyncDataStepAverageLatencyHelp = "Sync data average latency"
  - SyncFolderHierarchyStepAverageLatency = "Server-side synchronization: Sync Folder Hierarchy Average Latency"
  - SyncFolderHierarchyStepAverageLatencyBase = "Server-side synchronization: Sync Folder Hierarchy Average Latency Base"
  - SyncFolderHierarchyStepAverageLatencyBaseHelp = "Sync folder hierarchy average latency base"
  - SyncFolderHierarchyStepAverageLatencyHelp = "Sync Folder Hierarchy Average Latency"
  - SyncFolderItemsStepAverageLatency = "Server-side synchronization: Sync Folder Items Average Latency"
  - SyncFolderItemsStepAverageLatencyBase = "Server-side synchronization: Sync Folder Items Average Latency Base"
  - SyncFolderItemsStepAverageLatencyBaseHelp = "Sync folder items average latency base"
  - SyncFolderItemsStepAverageLatencyHelp = "Sync folder items average latency"
  - UpdateInExchangeStepAverageLatency = "Server-side synchronization: Update in Exchange Average Latency"
  - UpdateInExchangeStepAverageLatencyBase = "Server-side synchronization: Update in Exchange Average Latency Base"
  - UpdateInExchangeStepAverageLatencyBaseHelp = "Update in Exchange average latency base"
  - UpdateInExchangeStepAverageLatencyHelp = "Update in Exchange average latency"
  - UploadAttachmentsInExchangeStepAverageLatency = "Exchange Sync: Upload Attachments Average Latency"
  - UploadAttachmentsInExchangeStepAverageLatencyBase = "Exchange Sync: Upload Attachments Average Latency Base"
  - UploadAttachmentsInExchangeStepAverageLatencyBaseHelp = "Upload Attachments Average Latency Base"
  - UploadAttachmentsInExchangeStepAverageLatencyHelp = "Upload Attachments Average Latency"
  - WriteToCrmStepAverageLatency = "Server-side synchronization: Write to Crm Average Latency"
  - WriteToCrmStepAverageLatencyBase = "Server-side synchronization: Write to Crm Average Latency Base";
  - WriteToCrmStepAverageLatencyBaseHelp = "Write to CRM average latency base"
  - WriteToCrmStepAverageLatencyHelp = "Write to CRM average latency"

## Telemetry

Data is also pushed to telemetry using ETW tracing. Currently the following data is being sent to telemetry.

- 
- SSSVerboseTracingDiagnostics: Verbose tracing from server-side sync
  - SSSWarningTracingDiagnostics: Warning tracing from server-side sync
  - SSSErrorTracingDiagnostics: Error tracing from server-side sync
  - SSSTestExchangeConnectivityDiagnostics: Hybrid Exchange test connectivity data
  - SSSTestAndEnableMailboxDiagnostics: Test and enabled diagnostics data
  - SSSMailboxSyncCycleDiagnostics: Mailbox sync cycle diagnostics data