MAPILab Reports
Administrator guide
Document version 3.02

MAPILab Ltd, April 2010

Table of contents

1. Product ideology

MAPILab Reports for Exchange is a program which collects data about IT infrastructure, stores them in a database, and includes a large set of reports which allow IT personnel quick access to the information in a convenient form, without requiring special knowledge for interpretation.

There were four key concepts which drove the development of the Product:

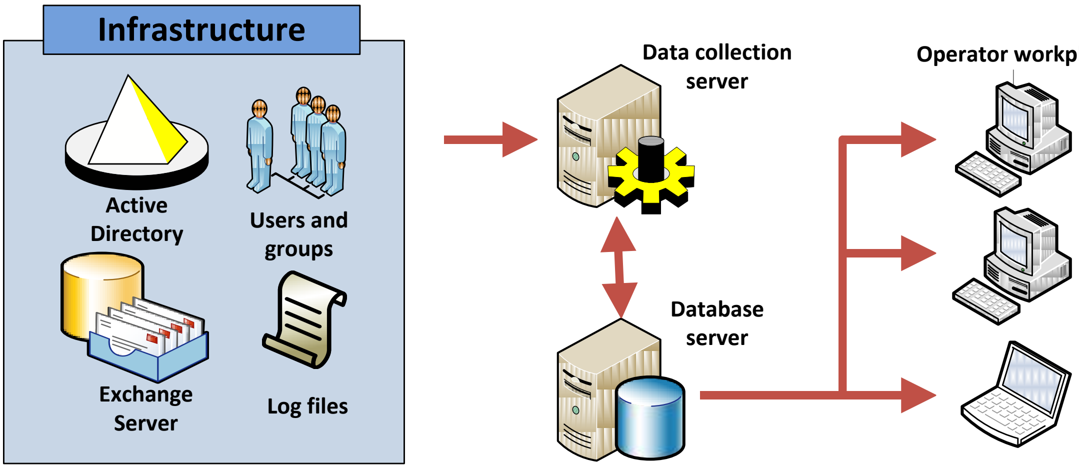
- **Quick report generation**. The User should not have to wait for the Product to collect data. Data should always be in the database, ready to be accessed. Moreover, some data can only be collected by the accumulation of statistical information over a given period of time—for example, data about growth trends in Exchange storages. A month could pass before you see some kind of trend in the graph of storage sizes. With this in mind, the Product is designed to collect data automatically, according to all infrastructure sectors which may be of interest (the User can, of course, limit the areas of data collection).

- **Simple Interpretation**. Some Users may not have high-level skills to interpret data, so the Product comes with a set of readymade reports which are designed with this in mind. Product reports do not contain 'raw' data—all information is displayed in a way that will not confuse the user. If some attribute or flag is obsolete, it will not be shown in the report, or it will be specially marked.

- **Simple Deployment**. MAPILab Reports for Exchange is not a solution which requires a complex installation process and specialists with advanced skills. It is a mass-market product which can be implemented by an average IT specialist without additional assistance. The entire Product is can be deployed on a regular workstation, and the product gathers all data remotely, without the installation of agents on any of the servers or workstations from which information is gathered, and without remote execution of code on these computers. These two advantages make deployment both simple and safe.

- **Extensive Infrastructure Analysis.** The most ambitious concept in the Product's design is the creation of a united informational and analytical center for all the IT infrastructure of an enterprise. This goal is reflected in the architecture of the Product through modules, that is, report packs including data collectors and report templates. The first edition of the Product contains two report packs: "Network infrastructure and Active Directory" and a pack for Microsoft Exchange Server. Additional report packs for Microsoft ISA Server, Microsoft SQL Server, and SharePoint are planned to be released subsequently. Users can license only those report packs which are of interest.

2. Architecture of the Product and general explanations



Here are the explanations of key concepts which are important to gain a general understanding of the architecture of the Product.

- **Infrastructure**. Active Directory domain controllers, workstations, Exchange servers and other **infrastructure objects**, from which MAPILab Reports for Exchange collects data. Note that data is remotely collected through various network interfaces, and no program modules are installed on infrastructure objects.

- **Administrator**. The User of the Product who performs installation, setup, and service of the Product.

- **Data collection server**. A physical or virtual computer, on which MAPILab Reports for Exchange is installed and from which data is collected. In other words, all program code which is run to perform remote data collection is executed on that computer. This computer has a **Console** installed, from which data collection can be set up and managed. This console can also be used to create reports. The Product license allows installation on only one data collection server (unless a special agreement is made).

- **Database**. MAPILab Reports for Exchange uses Microsoft SQL Server to store collected data. A license for Microsoft SQL Server must be acquired separately; it is not included in the Product. It is possible to use the free edition, Microsoft SQL Server Express. MAPILab Reports for Exchange does not put any restrictions on the location of the DBMS — it can be deployed on a data collection server or on any workstation or server on the organization’s network.

- **Operator**. A User who has rights only to generate reports using the data from the database.

- **Operator workstation**. This is the physical or virtual computer, on which MAPILab Reports for Exchange is installed but from which only reports may be generated from the database. Operators do not have the right to manage data collection or data collection settings. They only have the right to read data from the database and create reports. There is no limit on the number of operator workstations which can be created under a single product license.

Deployment options of MAPILab Reports for Exchange:

- **Compact deployment**. In this case, deployment is completed on a single workstation (for example, running Windows XP or Windows Vista), on which the data collection server is deployed and where the DBMS is installed. Users having the rights to interactive access to this computer and to launch components of MAPILab Reports for Exchange can manage data collection and create reports. For other users, automatic generation of reports according to the needs of individual users can be scheduled. Generated reports can be sent by email or published in a shared folder, on SharePoint, or on an FTP server.

- **Typical deployment**. In this case, the DBMS and the data collection server are located on different servers, and on one or more operator workplaces are set up. The reason for separation of the DMBS and the data collection server is usually to employ a high-performance database server which is already available in the organization, and to reduce the cost of licensing of the DMBS as well as the storage and data backup systems.

- **Mixed deployment.** For this type of deployment include, inter alia, the following options:
    - Distribution deployment. In some complex environments from a single server data collection may be physically inaccessible all the infrastructure objects of the organization. For example, when a branch of the organization. In this case, to collect data from remote branches with a central data collection server, such as the head office, is irrational. To organize the qualitative data collection, each network segment, an organization established its own data collection server, and all the data collection server stores data in one database.

    - Deployment with multiple databases. The product allows you to use different databases for different packages of reports and even for different categories of data. This type of deployment can be used for load-sharing and better management of access rights. Also in the case of the expected growth of the database to bypass the restrictions on free edition Microsoft SQL Server.

The license for MAPILab Reports for Exchange does not include a license for Microsoft SQL Server, and a license to MAPILab Reports for Exchange does not limit the number of operator workplaces. You can choose or change the type of installation according to your needs.

Technologically, for example, in terms of the integrity of the database for MAPILab Reports for Exchange, there is no difference if data collection is performed on one computer or multiple computers at the same time. However, it should be noted that data collection can cause a significant load on the infrastructure objects (for example, when collecting data on the contents of Exchange mailboxes), which may lead to lower performance, and even partial denial of service, if a resource-intensive task is launched simultaneously from multiple computers. Therefore, management of data collection and the task schedule should be entrusted only to qualified personnel. Also, please note that the license to use MAPILab Reports for Exchange provides for the use of only one data collection server in the organization. Therefore, if necessary, distributed deployment of the product, you must specify the appropriate number of licenses in our sales department.

3. Components

The Product consists of several components which can be divided into two groups: for users and for maintenance. A more detailed description of the user components is available in the **User Guide**; they include:

- **Console.** The console is a MMC3.0 snap-in and is designed to manage data collection, create the data collection task schedule, and analyze errors encountered in data collection and report creation. However, because of licensing and technical restrictions, these options are available only on the data collection server. On operator workplaces, operators can use the console only to create reports; an operator cannot manage data collection or set the data collection task schedule.

- **Report Viewer**. This component allows reports to be generated in many supported formats (XLS, HTML, PDF and others), but reports generated in the program's original format can have filters applied to them, as well as have the sorting or visual scheme changed.

- **Report Designer.** Allows the layout of reports to be changed, for example, deletion of unneeded columns or changing their width. This component is technically implemented as the part of Report Viewer and has no separate executable file.

- **Visual Scheme Editor**. Allows creation of your own visual schemes for reports using, for example, the company logo, corporate font and palette. This component is technically implemented as the part of Report Viewer and has no separate executable file.

Maintenance components:

- **Report packs**. These include software modules of **data collectors** (one or more) and report templates. Report packs must be installed both on data collection servers and operator workplaces. The product license allows installation of a report pack on one data collection server and an unlimited number of installations on operator workplaces. This version of MAPILab Reports for Exchange has in its composition of only one package report.

- **Utilities.** The Product comes with a set of support tools for maintenance.

4. Data collection

After product installation, its database is empty, and data must be collected before the first report can be generated. Typically, data collection is configured immediately after installation, and then the Product works in fully automatic mode. Thus, much attention must be given to the process of configuration, which includes four main steps:

- **Setting up the environment.** This step includes the creation of accounts for the data collection and the configuration of the infrastructure objects from which the product will collect data.

- **Setting up data collection.** Each report pack has a group of settings that allows control over both the scope of data collection and the parameters.

- **Setting up the schedule.** For automatic data collection, one or more data collecting tasks must be created with a schedule for their execution.

- **Diagnosis of malfunctions.** After the first collection of data, it is necessary to verify the results and identify possible errors.

All data collection is run from the data collection server, and all data is collected remotely using various protocols such as WMI, LDAP, WebDAV, HTTP, and others. No software modules are installed on infrastructure objects; and no code is executed remotely.

4.1. Environment setup

By the environment setting should understand the fulfillment of a number of actions to ensure the correct functioning of the program. Given the large number of data sources, as well as the variety used in data collection protocols, should be familiar with the principles of data collection products. This section carries information of exploratory nature, which may be needed only in case of non-standard version of the deployment of MAPILab Reports for Exchange or the inability to perform typical setup of the product.

For detailed instructions on setting up the typical deployment, including step by step instructions are provided in the **Installation Guide**.

Features of the program, namely, remote, agent less data collection implies privileges for such collection. Data can be collected in two ways: in the runtime data collection mode and through the data collection on schedule. We do not recommend using the first version of the data collection tasks, because execution time in a network of a large organization can be very long (up to several hours). This method can be used as a diagnostic, in the case of certain problems in data collection. In addition, data on the runtime data collection will be made with the right to run its user, which in most cases not enough for all categories of data. The main options for collecting data should be considered the tasks of data collection is automatically by schedule using Task Scheduler during the minimum load of information system of the organization. With this method of data collection ceases to play a role, as the time of the task, and that the necessary privileges - to data collection on a schedule using a specially created user accounts with the necessary high privileges.

To collect data uses the following data sources:

Active Directory - by protocol LDAP;

Exchange Tracking and Transport Agent Log - a direct reading of log files;

Exchange Management Shell;

Logs of IIS - direct reading files;

WMI - Remote reading certain sections;

IIS Metabase - remote connection to the database.

Data collection agent, launched as part of the data collection task on schedule, performed on behalf of the user specified in the settings of the data collection task. Therefore, the right to use the user must provide access to the facilities above.

Below is a table showing the use of user accounts privileges for the data collection by different data categories.

User Account for Data Collection recommended by the Installation Guide	**Data Category, collected on behalf of the user**	**In fact used privileges**
User 1: -Domain Admin -SQL db_reader -SQL db_writer*	Exchange: General information about organization	Reading all NC of Active Directory, it properties and permissions
		Reading IIS metabases of Exchange servers (LM/W3SVC)
		Reading mailbox and public folder stores using Power shell (Read, Read properties)
		Reading log files of IIS services of CAS-servers **
		Reading WMI namespaces root\CIMV2, root\MicrosoftExchangeV2 of Exchange servers
		Reading/Writing SQL database
	Exchange: Tracking logs	Reading NC и Configuration Active Directory
		Reading tracking logs, transport agents logs of Exchange servers

		Reading/Writing SQL database
User 2: - Exchange Organization Administrator (Exchange Full Administrator for Exchange 2003) - Receive-as on all mailbox stores - Local administrator of data collection server - SQL db_reader - SQL db_writer	Exchange: Mailbox content	Reading NC и Configuration Active Directory, it properties and permissions***
		Reading mailbox and public folder stores using Power shell (Receive-As)
		Reading WMI namespaces root\CIMV2, root\MicrosoftExchangeV2 of Exchange servers
		Reading IIS metabases of Exchange servers (LM/W3SVC)
		Reading stores through Exadmin of Exchange servers
		Reading/Writing SQL database
	Exchange: Public folders content	Reading NC и Configuration Active Directory, it properties and permissions
		Reading IIS metabases of Exchange servers (LM/W3SVC)
		Reading stores through Exadmin of Exchange servers
		Reading/Writing SQL database
	Exchange: Mailbox and public folders attachments	Reading NC и Configuration Active Directory, it properties and permissions
		Reading IIS metabases of Exchange servers (LM/W3SVC)
		Reading stores through Exadmin of Exchange servers
		Reading/Writing SQL database
	Exchange: Security settings	Reading NC и Configuration Active Directory, it properties and permissions
		Reading IIS metabases of Exchange servers (LM/W3SVC)
		Reading stores through Exadmin of

		Exchange servers
		Reading/Writing SQL database

* In most cases, domain administrators will have inherited rights to the database SQL, as well as the local administrator in the case for User 2 in a compact deployment of the product. Nevertheless, this situation is not always possible, and we recommend that you delegate the necessary permissions directly to user.

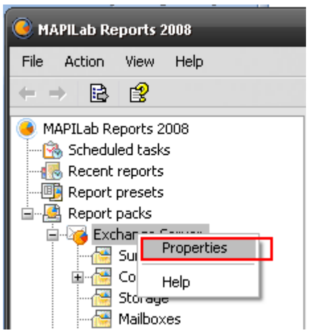
** for all data categories, which is used to collect data from the logs of services used by local administrator privileges, which have members of the Domain Admins group.

*** in most cases to obtain such data requires domain administrator rights. Since because of Exchange Server security policies, for domain administrators directly denied by reading the contents of mailboxes, the user, on whose behalf data collected, are not a member of the Domain Admins group. Obtaining the necessary data may come from the tables already filled with data collector of "Exchange: General information about the organization" category. This method is used automatically only if for data collector of "Exchange: Mailbox content" data category access to required information was denied. These methods are extended to other data categories.

Necessary to note that on the settings of the report pack you can specify settings to connect to SQL-server. If these settings are set to read and write data to the database will be used by the user account specified in these settings. This setting is convenient when the user running the **console** does not have sufficient rights to the SQL-server.

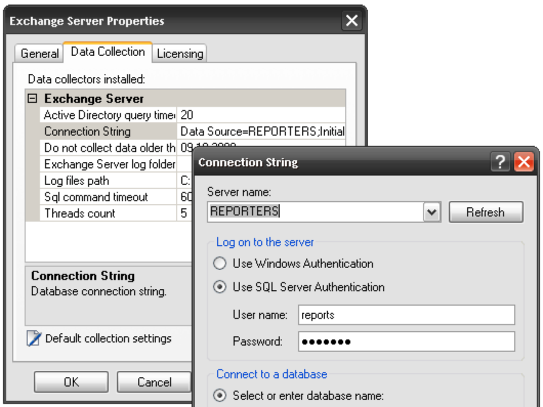
4.2. Report Pack settings

In the settings of the report pack is possible to change several default settings. In most cases, change the settings to be offered by default, not required.



4.2.1. Licensing setup

Setup for licensing is under the Licensing tab. During the trial period, MAPILab Reports for Exchange gathers data from all infrastructure objects within the limits of the data collection scope. After the trial period has expired, **license keys** must be entered in the settings of the report packs which correspond to the type and number of infrastructure objects from which data collection is performed. Further details on this subject can be found in the licensing terms
of the product and report packs.



Data collectors installed:

⊟ Exchange Server	
Active Directory query time	20
Connection String	Data Source=REPORTERS;Initial
Do not collect data older th	09
Exchange Server log folder	
Log files path	C:
Sql command timeout	60
Threads count	5

Connection String
Database connection string.

Default collection settings

Server name: REPORTERS

Log on to the server

○ Use Windows Authentication

● Use SQL Server Authentication

User name: reports

Password: ●●●●●●●

Connect to a database

● Select or enter database name:

4.2.2. Connection to the database

The database connection settings are set-up on the 'Data Collection' tab and include

“Connection string” and a “SQL command timeout”. MAPILab Reports for Exchange can save data to a single database, or to separate databases, depending on the number of **data collectors** included in the given report pack.

This feature can be used to avoid the limitation of the maximum database size (e.g. Microsoft SQL Server 2005 Express Edition database is limited by 4 GB) or, in case of increased load, spread the database out over different physical servers.

Before you change the ‘Connection string’ in the settings, you must create the database and initialize it, creating the structure of tables in the database. This is most easily done by launching the MAPILab Reports for Exchange **installation program** and leaving only the **database components** for the needed report packs in the list of installed components. The database is created and initialized as part of the process of installation. This process can be repeated several times, depending on the number of databases which you want to create.

Attention is drawn to configure authentication settings for connection to SQL-server. Use this setting determines some features of the product. You can specify two types of authentication: Integrated Windows and SQL Server. Authentication type is exhibited when you install the product on the basis of the data specified in the Setup Wizard product. It does affect the data collection tasks, runtime data collection and generating reports. For example, if you check a built-in Windows authentication database connection for data collection tasks, runtime data collection and in the generating of reports will be based on user credentials, under which the process is running. I.e. the report generation and runtime collection – on behalf of user which runs the MMC, when the work data collection tasks - on behalf of user which credential set in settings of data collection task. When using SQL authentication, all connections to the database will be made with the use of credentials of the user specified in the connection string settings, regardless of who started the process. This feature is necessary to consider adjusting the operator workplaces and performing runtime data collection.

Also, for all report packs there is a “SQL command timeout” setting, which determines the timeout of the SQL query in seconds. A more detailed description of this setting can be found in the document

http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.commandtimeout(VS.80).aspx

and other Microsoft documents.

4.2.3. Log files path

This setting is found in the “Data collection” tab – “Log files path”. This setting has a value “%AllUsersProfile%\ MAPILab Ltd \ MAPILab Reports \ Logs" by default. To turn off logging, set the value to an empty string. Use of the log folders is explained in detail in section 4.4 (Diagnosis of malfunctions).

4.2.4. Threads count

This setting is present in all report packs, and is available for all **data collectors**. A good example of the use of this setting is data collection on IIS logs. A data collector must, in such cases, connect to each server. The number of threads in this case indicates the number of servers to which the data collector is allowed to connect simultaneously. An increase of the number of threads of ten times allows data to be collected ten times faster.

However, each thread requires memory and CPU resources of the data collection server and the database server. If too many threads are run simultaneously, each new thread will produce a decline in performance, since, for example, the operating system will have to use a paging file for the data collection process and other processes—this will reduce performance.

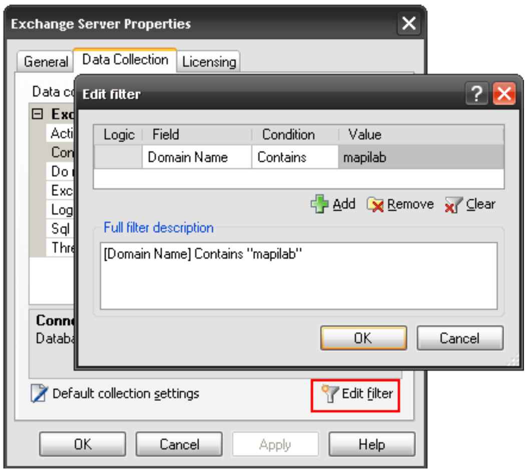
It is difficult to make a concrete recommendation concerning the ideal number of data collection threads, since it is very dependent on the **data category** which the collector is gathering, the hardware resources of the data collection server, the type of the given deployment of the Product, and even on the practical use of the Product in the organization.

Therefore, the general recommendation is to find the golden middle. If data collection requires an unacceptably long time, the number of threads should be increased and then the change in performance in the data collection server and the database server should be evaluated.

Aside from the number of data collection threads, performance of the Product is significantly affected by the **data collection schedule**. You can reduce the total data collection time not only by increasing the number of threads, but also by simultaneous execution of multiple **data collecting tasks** (see pt. 4.3).

4.2.5. Data collection filters

This is one of the most important settings because it limits the scope of data collection. In the "Data Collection" tab, press the "Edit Filter" button to open the settings dialog box.



Logic	Field	Condition	Value
	Domain Name	Contains	mapilab

Full filter description

[Domain Name] Contains "mapilab"

The simplest way to discuss the action of filters is by example. Suppose that we are launching the Product at a branch of the organization, and we need reports and we need only the data from servers located in the branch, and included in a specified administrative group. Suppose further that we physically do not have access to servers at the head office or other branches, and an attempt to collect data from them is bound to be unsuccessful, since connection to each server is required. In this case we can use a collection filter to limit data collection specifically to this administrative group. The data collectors will not try to connect to objects which are excluded by the collection filters.

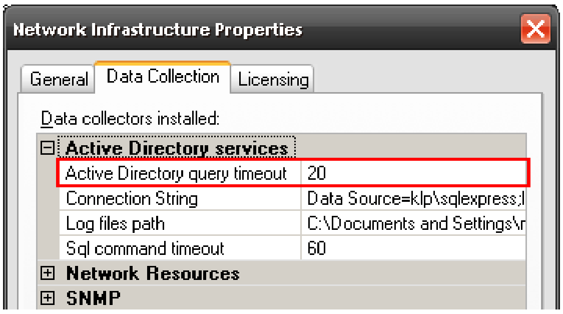
However, in some reports, for example *Exchange Servers*, information about computers which are 'excluded' does appear. This is not an error. The data for this report are taken from Active Directory, or more exactly, from the domain controller, without connection to the computers which are filtered out. This is why a data collection filter by administrative group does not apply in this case. Thus, since the domain controller contains data from all computers in the domain, this report will include information on computers from the head office and the other branches.

In order to exclude data about computers from the head office or other branches in *Exchange Servers*, a data filter for administrative groups can be applied during the generation of the report (see "Creating Reports" in the **User Guide**).

Data collection filters can also be extremely useful in the diagnosis of errors (Section 4.4), functioning to restrict data collection to those objects which are being investigated, shortening the time needed for data collection and consequently reducing the time needed to correct the error.

4.2.6. Other settings of data collectors

Aside from settings listed in the previous points of this section, have additional settings, which are assigned in the "Data Collection" tab.



Data collectors installed:

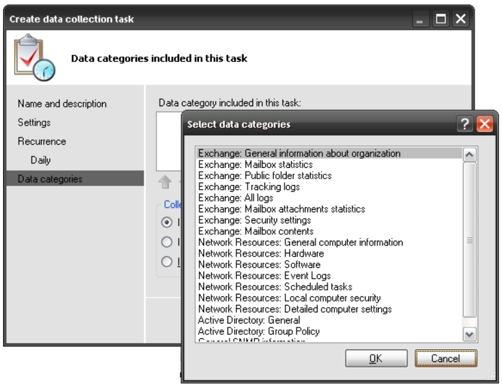
⊟ **Active Directory services**	
Active Directory query timeout	20
Connection String	Data Source=klp\sqlexpress;I
Log files path	C:\Documents and Settings\r
Sql command timeout	60
⊞ **Network Resources**	
⊞ **SNMP**	

Active Directory query timeout. The value is in seconds, limiting the maximum period of waiting for a response from the Active Directory service.

Do not collect data older than. Date of limiting the processing of records from the Exchange log files, including message tracking logs, transport agent logs), as well as OWA / IIS service logs. In general, the setting is used only once, limiting the first data collection, and billed automatically when you install the product. Change this setting only if you need to collect data of past events. Note that such data tend to be incomplete.

Log files path. List of network paths to the log folders of Exchange servers, an optional parameter. Name or IP-address of the server in the network path must match the name or address of Exchange server; otherwise such a path will be ignored. If in this setting no path are defined for each server, the processing of its log data collector determines the local path of logs from the server settings (C: \ Log files) and creates a network path through the administrative shares (\ \ servername \ C $ \ Log files). For a one server can be defined several different paths - the tracking logs path, transport agent log path and OWA / IIS log path. This setting can be used if you can't uses domain administrator user account for data collection in "Exchange: Exchange tracking logs" category. Also, this option should be used to gather information from the Exchange Edge Transport Server, because as a rule, edge servers are made in the demilitarized zone and access there of data collection server will be closed. In the case of collecting data from the Edge Transport Server, you must open the network access of data collection server to edge on ports 445 and 139 TCP, and also define in this setting the path to the log files of Edge Transport Server. For example: \ \ EDGE \ Logs \ AgentLog; \ \ EDGE \ Logs \ MessageTracking. It is also necessary to ensure that the user, on whose behalf the data collection is performed, have permission to read this file share.

4.3. Data collection tasks and data categories


Create data collection task
Data categories included in this task
Name and description
Settings
Recurrence
Daily
Data categories
Data category included in this task:
Select data categories
OK
Cancel

Exchange: General information about organization
Exchange: Mailbox statistics
Exchange: Public folder statistics
Exchange: Tracking logs
Exchange: All logs
Exchange: Mailbox attachments statistics
Exchange: Security settings
Exchange: Mailbox contents
Network Resources: General computer information
Network Resources: Hardware
Network Resources: Software
Network Resources: Event Logs
Network Resources: Scheduled tasks
Network Resources: Local computer security
Network Resources: Detailed computer settings
Active Directory: General
Active Directory: Group Policy

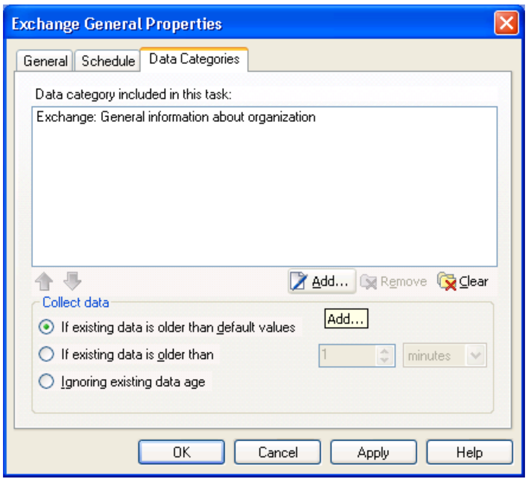
A **data collection task** is technically a task for the Windows scheduler. During the creation of a task, **data categories** are specified, for example, “Exchange: Mailbox Contents” and “Exchange: Security settings”. Data collection will be performed with the rights of the user account specified during the creation of the task. Account name and password are not saved in MAPILab Reports for Exchange; they are passed to the Windows Scheduler during the creation of the task. A complete list of data categories and rights for data collection is attached as pt. 4.1 of this guide; step-by-step instructions for the creation of accounts with the required rights are given in the **Installation Guide**.

In most situations, simply looking at the names of the data categories and reports is sufficient to determine which reports belong to which categories. Thus, collection of data of the category “Exchange: mailbox contents” provides the data for *MIME-types of attachments in mailboxes* report, *Mailboxes and subfolders* report, and so on.

It is a good practice to plan resource-intensive tasks, like the collection of “Exchange: mailbox contents” data category, at a time when the utilization of the corresponding service is minimal—for example, at night or on weekends. If you add more than one **data category** to the data collection task, the data for each category is gathered in its turn. Management of the order of data collection is accessible using up and down arrows under the list of data categories in the properties of the task.

If we have performance headroom, but limited time in which to complete data collection, it is possible to create several data collection tasks which are executed simultaneously and gather data of separate categories.

Using several simultaneous data collection tasks which gather data of a single category is not appropriate and in many cases can reduce efficiency. To speed up data collection for a given data category, you should increase the number of data collection threads for the **data collector** (see Point 4.2.4).



Data category included in this task:
Exchange: General information about organization

● If existing data is older than default values
○ If existing data is older than 1 minutes
○ Ignoring existing data age

During or after the creation of a data collection task, you can, in the data collection tasks properties, manage data collection with consideration of data obsolescence (see screenshot). For an understanding of these settings, see section 5, “Storage and selection of data”, and the detailed explanation in pt. 5.3.

4.3.1. Journal of data collection tasks

This setting is assigned in the *Log settings* tab in the main settings window of MAPILab Reports for Exchange. To open a window in the console tree, right-click on the MAPILab Reports node and select *Properties* from the context menu.

Default value is "%AllUsersProfile%\Application Data\MAPILab Ltd\MAPILab Reports\Logs". To disable logging, specify an empty string as the value. Working with log files is described in detail in Section 4.4 (Diagnosis of malfunctions).

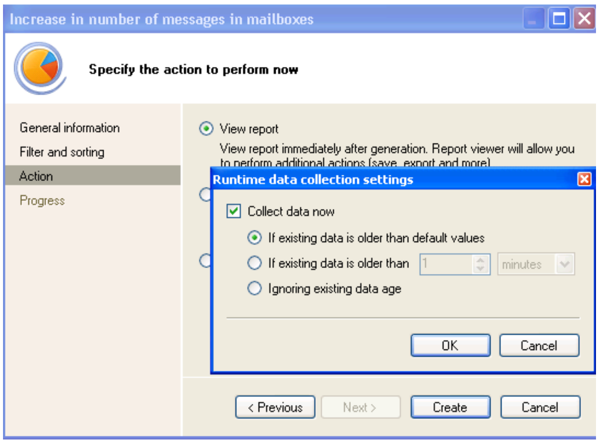
4.3.2. Runtime collection mode and data category exceptions

Although the relationship between data categories and reports is almost always clear from their respective names, there are exceptions.

It consists, it is that a number of reports require data not from one category, but from several. Some consolidated report about storages may require data both from public folders and from mailboxes.

On the Product website there are documents for all report packs which show the relationships not only between reports and data categories, but also their relationship with tables of the database.

When creating any report on the data collection server, the data for the report can be **runtime collected**. One of the windows of the report wizard allows the administrator to choose this mode (see the **User Guide**). In this mode, data collection is not made according to the data categories which are used in the data collection tasks. Rather, only those data which are needed for the specific report are collected, and only the tables relevant to that report are updated.



● View report

View report immediately after generation. Report viewer will allow you to perform additional actions (save, export and more)

Runtime data collection settings

☑ Collect data now

● If existing data is older than default values

○ If existing data is older than 1 minutes

○ Ignoring existing data age

Usually the **runtime collection mode** is not used, and reports are created using those data which are already in the database and were collected by scheduled data collection tasks. Moreover, these practice not recommended for two important reasons. First, data collection for some reports may take several hours and affect system performance. Second, this mode uses the privileges of the user who runs the report (and not the privileges of the account used in data collection tasks), which forces the administrator to work with high-level privileges. Of course, this option is not technically possible at **operator workplaces**.

But if is it is necessary to bypass an exceptional situation, as in the case to obtain operational data, the administrator can launch the report wizard, turn on runtime collection mode for the report, and save the **report preset** (see **User Guide**). After this, create a **report generation task** in the scheduled tasks using this report preset.

4.4. Diagnosis of malfunctions

After the first collection of data is completed, a diagnostic check for malfunctions must be performed.

The first step in the diagnostic process is to check the status of completion of **data collection tasks** in the 'Scheduled tasks' section of the console. When a task in the list is selected, the region below the list shows its current status, the time when it was most recently run and its data collection results. As a rule, all arising from errors do not cause problems with interpretation. If you can't correct the error yourself, you should contact the technical support service.

Some errors in data collection are temporary in nature, caused by the temporary unavailability of infrastructure objects to the data collector. Other errors are permanent, caused by the lack of access rights to data collection or the absence of services needed for data collection on infrastructure objects. In the section, "Solving typical problems" of the **Installation Guide**, a list of the most typical problems and methods for solving them is given. The purpose of that section is to provide general methods for solving these problems.

The second stage of diagnosis is to run the Product's reports with the goal of verifying that data is indeed collected from all intended infrastructure objects.

If incomplete or missing data is detected in any report, it is necessary to determine the reason and code of the error. To do this, open the corresponding report log file. To determine the location of the log file, see Point 4.3.1.

If you can't identify the report's corresponding log file using the name of the report and the name of the log file, determine the names of the tables in the database which are used in the report (see Point 4.3.2). The name of the log file is fully consistent with the name of the table. More than one table may be used to create a report, but some of them are index tables for which log files are not kept. Some reports which include data of different categories may correspond to several log files, but knowing what data is unavailable, you can immediately find the needed file.

After identifying and removing an error (see also **"Solving typical problems"** in the **Installation Guide**), data must be collected again to make sure that the error has indeed been fixed.

In this case, it is very useful to use the **runtime data collection mode** (point 4.3.2), since this provides a significant reduction in data collection time, gathering data only for those tables which are needed to create the report. A second way to reduce time of data collection is to assign additional limits using data collection filters (point 4.2.5) that shrink the scope of data collection during testing. A third method is to increase the number of data collection threads allowed to the data collector (point 4.2.4).

5. Storage and selection of data

5.1. Historical and statistical reports

Each report pack has its own set of historical and statistical reports. Generally, statistical reports show changes in one or another value over time, and historical reports show a snapshot of a group of values. To better understand the difference, two examples follow.

A good example of a statistical report is a report of the growth of an Exchange server storage file. This report contains, among other things, a graph showing the size of the file on a daily or monthly basis. When creating the report, the user gives a range of dates specifying the period which the report should cover.

An example of the historical report can provide a report on access permissions of mailboxes. It contains a table with a list of mailboxes and user rights. As the product keeps a complete history, a user can build a report for any date, to find permission for a separate box by today or what permissions were a year ago. When creating this type of report, the user gives a date, called an **actual point**, for which he wants to receive a report.

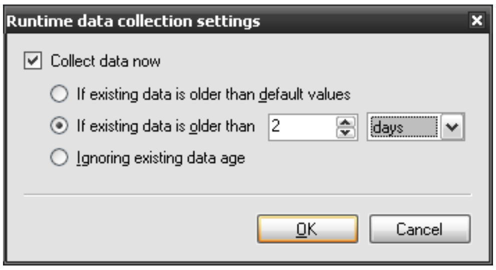
5.2. How data are stored in the database

Aside from special cases, the database uses the same method to accumulate data for both statistical and historical reports. As soon as a value or group of values changes (which is found out by regularly scheduled data collection), a new record with the value and the timestamp appears. For example, for a report that shows the growth of Exchange server storage on a daily basis, data is not saved on each calendar day. Only days when a change in the value is observed are saved.

Data collection is, therefore, a little more complicated than it could be—the data collector does not simply read the value of the object and write it in the database, but checks whether the value has changed in comparison to what is currently in the database. If the data have not changed, then the record and its timestamp to not change, but a special marker indicating the time of the most recent confirmation is set. This reduces the size of the database considerably, without losing data. The supposition is made that data collection occurs often enough, that loss of intermediate values are insignificant.

Exceptions to this principle of data storage are the summary tables, for example, containing records with monthly values of some magnitude. They serve two purposes. First, calculation of these values each time while forming a report may take too long. Secondly, with consideration that the size of the database must be reasonable, data in tables of this type can be saved much longer, than a data in tables containing hourly values, granting users, for example, detailed statistics for the last 3 months and a summary of the last five years.

5.3. Data obsolescence



☑ Collect data now

○ If existing data is older than default values

● If existing data is older than 2 days

○ Ignoring existing data age

In practice, there are situations when the actual points are not enough to create a historical report.

Therefore during the creation of a report, the user can choose one of three options in the settings window of the report wizard (see screenshot):

- **Use default settings**. In other words, rely on the intellect of the program to decide which data in the database are obsolete, and which are not. The program has various internal settings to determine obsolescence for different data.

- **Return all database data.** The report will contain all data, independent of the last time of collection or confirmation.

- **Return data not older than [time parameter].** In this case, the report will include all data that is was collected or confirmed no more than an entered value of minutes, hours or days before the point of relevance.

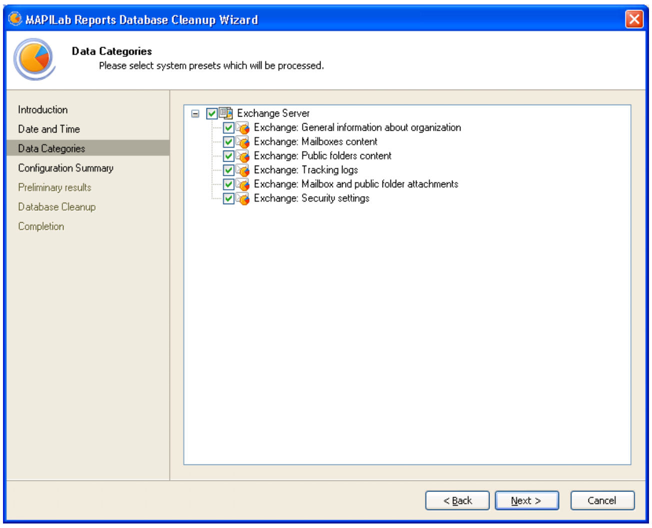
Obsolescence as a concept is not only used in data selection, but also in data collection. The settings for obsolescence for data collection can be defined in the settings for **data collection tasks** (point 4.2.6). Understanding the how these settings work has great practical value.

The settings for obsolescence for data collection also allow optimization and acceleration of the collection process. For example, when collecting data on the Exchange tracking logs will be made reading and retrieving data for the entire period of time, which is stored in logs. This may be very long time, so it is reasonable to collect only new data, those are younger than records in the database. This option of obsolescence settings for data collection exposed by default on creation of the data collection tasks.

5.4. Removing old data from the database

To remove old data from the database, as well as to learn the size of data of different categories in the database, you can use the utilities supplied with the Product: DatabaseUtil.exe and DatabaseUtilCon.exe, which are in the folder where the Product is installed. Both utilities perform the same function; the first provides a graphical user interface (a wizard), and the second - a command-line interface.

When running DatabaseUtil.exe, in the first step you choose a master date—data older than this date will be deleted. You can then select categories of data to be cleared, and in the third step of the wizard you are able to turn on advanced mode to set your deletion preferences at the lowest level - in tables. Before starting the deleting procedure, the wizard will show the amount of data that will be removed and will propose to shrink the database after completion of data removal.



Data Categories
Please select system presets which will be processed.

- ☑ Exchange Server
  - ☑ Exchange: General information about organization
  - ☑ Exchange: Mailboxes content
  - ☑ Exchange: Public folders content
  - ☑ Exchange: Tracking logs
  - ☑ Exchange: Mailbox and public folder attachments
  - ☑ Exchange: Security settings

To delete old data by a schedule, you can create a task in the Windows scheduler, which will use the DatabaseUtilCon.exe. Run the program without parameters to get help concerning its command line.

6. Answers to FAQ’s

6.1. How can data be collected faster?

Increasing data collection speed may be achievable by increasing the number of data collection threads for the data collector, simultaneous use of more than one data collection task and competent management of data obsolescence settings. The exact answer depends on exact what you want to accelerate: the collection of data of a particular data category, the overall speed of the entire data collection process, or the updating of a particular data category. A complete answer can be acquired from points 4.2, 4.3 of this guide.

6.2. How can the Product be deployed in a company with branch offices?

For this, the Product must be installed in each separate branch location, limiting data collection with the use of data collection filters (see point 4.2.5). It can be used a separate database in each branch or to collect data in one central database. It should be borne in mind that in the first case, producing a consolidated report including data from the central office and branches is not possible using the Product by itself. A consolidated report can only be created by uniting the data of several reports in a third-party product, such as Microsoft Excel. In the second case, you should restrict access to the database for users of branches, as they will be able to get reports on the organization as a whole.

Although the Product’s architecture provides for the potential of consolidation of multiple databases, this feature is not yet developed, and its viability is being studied. The main complication lies in the need to replicate databases, the size and growth of which is very significant.

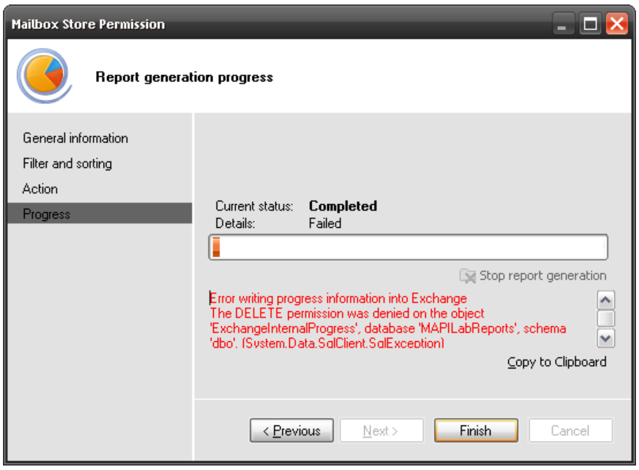
An alternative to consolidation of databases could be the possibility to create reports from data drawn simultaneously from multiple databases.

We would be glad to hear your opinion concerning this issue.

6.3. Can two data collection servers be used?

A product license allows use of only one data collection server. Contact our sales department and describe the reason that makes this situation imperative. In many cases we may expand a license without additional payment.

6.5. How can a single operator’s access level be limited to selected reports?



Current status: **Completed**
Details: Failed

Error writing progress information into Exchange
The DELETE permission was denied on the object 'ExchangeInternalProgress', database 'MAPILabReports', schema 'dbo'. (System.Data.SqlClient.SqlException)

There are two solutions to this issue which differ in simplicity of execution and reliability.

First solution – delete the report template files (files with the MPRT extension), which are stored in the subfolders of the Product installation folder. After deleting the report template file, the report will not show up on the list on the console. However, physical access to the data in the database still exists in this case, and this method is flawed in terms of security. Additionally, even if this solution is for some reason preferable, the process will have to be repeated after every update of the Product.

The second solution is to limit access to corresponding tables in the database on the level of the database server. This solution is more complicated, but completely sound from a security point of view and does not need to be repeated after product updates. How to determine the relationship between reports and tables is stated in Point 4.3. In implementing this solution, there is no need to deny access to all tables which are used to create a report. It is sufficient to deny access to one table containing critical data. When an operator launches a report which requires a table which is not accessible, there will be a message about an access error and the report will not be formed.

The two solutions, of course, can be combined.

6.5. Does the Product support Exchange 2007 Edge Transport servers?

Yes. MAPILab Reports for Exchange support data collection from Exchange Edge Transport Servers. Additionally, all kind of high availability environments are supported, including Clustered Mailbox Servers and Network Load Balanced Client Access Servers. Data collection from the Unified Messaging Servers is unsupported.