
The Mainframe Software Partner
For The Next 50 Years

# File-AID/EX

# Best Practices

**Release 19.01**

Please direct questions about File-AID/EX or comments on this document to:

**Compuware Customer Support**

**https://go.compuware.com**



Doc. JUL2018
June 11, 2018

# Contents

# Introduction

# About This Guide

This guide provides the information needed to install File-AID/EX. It consists of the following chapters and appendixes:

- Chapter 1, “Related Extract Performance Tuning” provides File-AID/EX performance tuning tips for Related Extract.
- Chapter 2, “File-AID/EX Related Extract/Related Loader Benchmarks” provides recommendations for improving File-AID/EX Related Extract and Related Loader performance while using SQL Server.

## Who Should Read This Guide

This guide is written for developers, testers, and database administrators who will using or maintaining File-AID/EX.

This guide assumes that users are already familiar with Windows.

## Conventions Used in This Guide

The following conventions are used to draw your attention to special information:

| This convention | Identifies |
|---|---|
| **boldface** | Information that you type, choices that you select from a window or menu, and keys that you press. Information should be typed in lowercase letters unless otherwise indicated. Boldface is also used to emphasize important points. |
| <*italic*> | Placeholders for items you must supply. For example: when the guide says to type <*drive*>:\, type the letter of the drive followed by a colon and a slash. Information should be typed in lowercase letters, unless otherwise indicated. Italic is also used to introduce new terms and indicate book titles. |
| **Note:** | Information that emphasizes important points. |
| **CAUTION:** | Information to prevent data loss or corruption. |

# Documentation

If you cannot locate the information you need or the information in this guide is not clear, please let us know.

### Publications

The File-AID/EX documentation set includes the following references:

- *File-AID/EX Installation Guide* includes system requirements and instructions for installing File-AID/EX and File-AID/EX Enterprise Edition. This guide is provided in PDF format.
- *File-AID/EX Getting Started* includes an overview of File-AID/EX and tutorials that teach you how to use the File-AID/EX component. This guide is provided in PDF format.
- *File-AID/EX Best Practices Guide* includes information on configuring File-AID/EX and benchmarks to help you make the best use of File-AID/EX.
- File-AID/EX online help provides descriptions of the File-AID/EX tools, operating procedures, and reference information. On the **Help** menu:
  - Click **Contents** to view an outline of available topics. This provides information in a logical order to help you perform tasks in the order they need to be performed.
  - Click **Index** and type the term for which you seek information.
  - Click **Find** to search for words and phrases in help topics. This will usually offer several topics from which to choose. Click a topic to select it.
  - Click **Help** on an active window for specific help with that feature.

### Viewing the Online Books

File-AID/EX online books are provided in PDF format, and require Adobe Reader 6.0 or more current to view them. The free Adobe Reader is available on the Adobe web site at www.adobe.com.

The online books are available from the Compuware Support Center located at https://go.compuware.com/.

## Product Support

Visit the Compuware Support Center, **https://go.compuware.com**, to find product documentation, knowledge articles, and other technical resources. You can open a case with the Customer Solutions team, order products, and much more.

Contact Customer Solutions by phone:

- USA and Canada: 1-800-538-7822 or 1-313-227-5444.
- All other countries: Contact your local Compuware office. Contact information is available at **https://go.compuware.com**.

Visit Compuware on the web at **http://www.compuware.com** for additional product information.

# Chapter 1.
# Related Extract Performance Tuning

Tuning a related extract to achieve maximum performance requires knowledge of very technical details that affect an extract. This document attempts to strike a balance between too much detail and too little, erring on the side of too much so technical users are able to get the most out of Related Extract.

In File-AID/EX Release 5.2, a new Related Extract engine, the Database Server-Driven Engine, was created that provides significantly better extract performance in most cases. This Database Server-Driven Engine currently supports MS SQL Server and Oracle databases only. Support for DB2 and Sybase is expected to be added in a future release. The original engine, the Execution Server-Driven Engine, that was used in previous File-AID releases is still available.

Following is an explanation of the available tune-up options for both engines and general recommendations for performance optimization.

## Typical Network Deployment

To run a related extract with fast performance, the network configuration must be taken into account. There are three major network components for Related Extract:

1. **Client Machine:** The machine that the user interface (such as the Related Extract Wizard or ConverterPro) is run on.
2. **Execution Server Machine:** The machine that does the work, including accessing the database and saving the data to disk.
3. **Database Machine:** The machine that houses the relational database, such as DB2, Oracle, SQL Server, or Sybase.

**Note:** It is possible to run all three components on the same machine, on three separate machines, or some other combination of machines. Since this choice can have a drastic impact on the performance of an extract, a simple rule of thumb is suggested: each component should be located close enough to the other components that it runs quickly enough, but not so close that it contends for the same resources and runs too slowly.

The application of this rule is explained in the following sections.

### Client Machine

The client machine's job is just to gather the work specification and send it to the execution server for processing, and to summarize its progress during processing. Since this does not require a fast data connection to the execution server, it is typically located on the user's local PC.

It is even possible to run the client with only a WAN connection to the execution server. However, the exact bandwidth required for the client has not been benchmarked; its ability to run over a WAN is provided as an example of what various customers have found successful. Before deploying a client using a WAN connection to the execution server, be sure to run tests of typical usage that verify the connection is fast enough.

## Execution Server Machine

The execution server machine does most of the work in a related extract and requires a fast connection to the database. As explained above, it requires a relatively low speed connection to the client.

The execution server makes intensive use of RAM and disk in order to perform a related extract. Therefore, if the execution server is run on the same machine as the database, they will likely contend for both RAM and disk and performance will suffer. Because of this, it is often better to run the execution server on a machine separate from the database but with a fast network connection between the two.

It should be noted that for small databases, the execution server can run on the same machine even though resources are shared between the two. How small is small enough? The general rule is if the database can satisfy the requests of all of its users while running a related extract using a local execution server, then it is small enough to be on the same machine.

## Database Machine

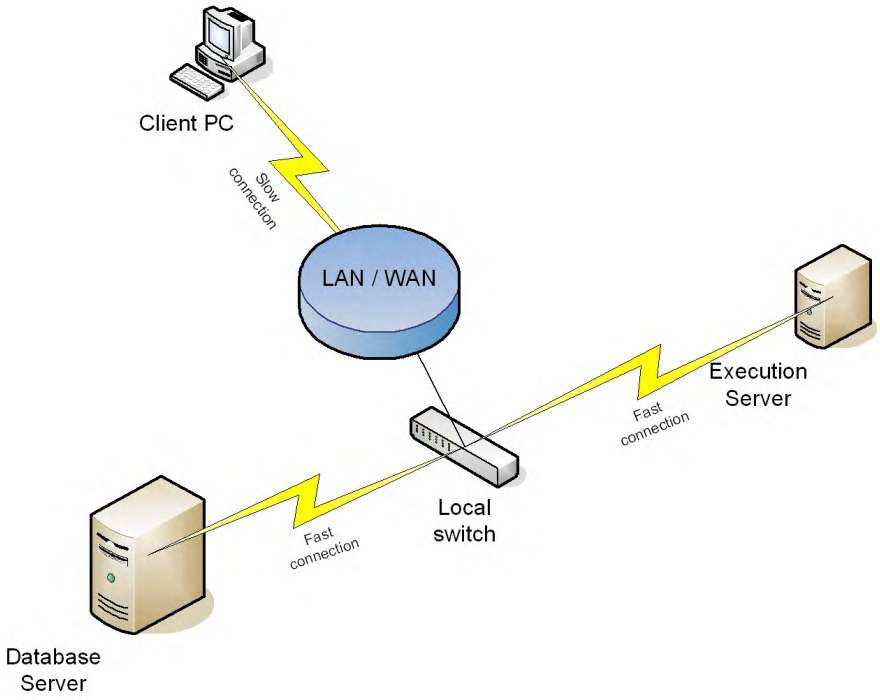
The database supplies the data to be extracted and therefore requires a fast connection to the execution server. However, large database machines are often treated as critical resources and installing the execution server on such a machine is typically not allowed. And as mentioned above, when the execution server shares resources on the same machine as the database, the performance of both often suffers.

Therefore, it is recommended that the execution server be placed on a machine separate from the database with a fast network connection between the two. However, for small databases it is possible to run the execution server and the database on the same machine.

## Example Configuration

The following network diagram shows the simplified network configuration of the server machines used for File-AID/EX performance benchmarks. The execution server and database server connect through the same network switch for speeds sake. The client PC that displays the user interface is connected through a slower network connection as described in “Client Machine” above.

**Figure 1-1.** Simplified Network Diagram — Related Extract




A basic description of the execution server and database server hardware used for File-AID/EX performance benchmarking is provided in the table below.

**Table 1-1.** Execution Server and Database Server Hardware Requirements

| | **EXECUTION SERVER** | | **DATABASE SERVER** |
|---|---|---|---|
| **HARDWARE** | | | |
| CPU | Intel Xeon x5355<br>8 cores @ 2.66 GHz | | Intel Xeon x5355<br>8 cores @ 2.66 GHz |
| Memory Model | 64-Bit | | 64-Bit |
| Virtual Storage | RAM - 16 GB<br>Pagefile - 24 GB | | RAM - 16 GB<br>Pagefile - 24 GB |
| Hard Disk Drive | C: 19.6 GB<br>D: 499.0 GB<br>S: 25.3 GB (Pagefile) | | C: 20.5 GB<br>D: 1.31 TB<br>S: 30.2 GB (Pagefile) |
| | | | |
| **SOFTWARE** | | | |
| Operating System | Windows Server 2003<br>.Net 3.5 SP1 | | Windows Server 2003<br>.Net 3.5 SP1 |
| File-AID/EX | 4.3 | | |
| Oracle | | | 11g |
| SQL Server 2008 | | | 10.0.1600 |

# Performance Tuning for Release 5.2 and More Current

The new Related Extract engine, the Database Server-Driven Engine, became available with File-AID/EX Release 5.2. The Database Server-Driven Engine more deeply utilizes the relational DBMS for analysis of relationships and selection of data to be extracted. In

many cases, it provides over 10 times speed improvement compared to the Execution Server-Driven Engine.

The Database Server-Driven Engine is currently offered only for Microsoft SQL Server and Oracle. The extraction engine can be selected on the **Execution Options** dialog box's **Advanced Settings** tab during creation or modification of an extract specification.

## Data Preparation and Data Processing

During the new extract process, the engine breaks the extraction job into two smaller pieces: data preparation and data processing.

**Data preparation** tasks generally involve the execution of SQL queries on the database server that includes creation (and deletion) of tables for temporary storage of data. Depending on the amount of data to be analyzed and extracted and the complexity of relationships involved, these tasks may require significant resources (primarily CPU time and available physical memory). A user can regulate usage of the database server resources by adjusting the data preparation threads setting (see "Advanced Settings" on page 10). Typically, data preparation tasks do not put any significant load on the File-AID execution server.

**Data processing** tasks use only a very low load on the database server but they may require significant resources on the execution server as well as disk drives or file servers that the extracted files are written to. This primarily depends on the amount and type of data to be read and data privacy rules to be applied to the extracted data. A user can adjust the load on these resources by changing the data processing threads setting.

The main goal of performance optimization for the Database Server-Driven Engine is to find a balance between the execution of data preparation and data processing tasks in a way that utilizes all available resources on the database server and execution server without overloading them. Some tips and hints on how to establish such a balance are given in the following sections.

## Advanced Settings

The Database Server-Driven Engine provides three advanced options that can be configured on the **Advanced Settings** tab in the **Execution Options** dialog box.

The **Data Preparation Threads** option specifies the number of data preparation tasks that can be executed in parallel. Available values include the following:

- **Auto:** Allows the execution server to determine the ideal number of threads. This option provides acceptable behavior for most extracts that do not involve extensive data or relationships.
- **Server Setting:** Directs the execution server to use the **Extract_Max_Db_Threads** value defined in the engine.properties file. In Release 5.3, this file is located by default at \ProgramData\Compuware\FAEX\Cfg (Windows Vista and later) or at \Documents and Settings\All Users\Compuware\FAEX\Cfg (prior to Windows Vista). In Release 5.2, this file is located by default at <*install folder*>/cfg.
- **Custom:** Directs the execution server to use the value specified by the user.

The **Data Processing Threads** option specifies the number of data preparation tasks that can be executed in parallel. Available values include the following:

- **Auto:** Allows the execution server to determine the ideal number of threads. This option provides acceptable behavior for most extracts that do not involve extensive data or relationships.
- **Server Setting:** Directs the execution server to use the **Extract_Max_Threads** value defined in the engine.properties file. In Release 5.3, this file is located by default at \ProgramData\Compuware\FAEX\Cfg (Windows Vista and later) or at \Documents

and Settings\All Users\Compuware\FAEX\Cfg (prior to Windows Vista). In Release 5.2, this file is located by default at <*install folder*>/cfg.

- **Custom:** Directs the execution server to use the value specified by the user.

The **Schema to use for temporary tables** option specifies a schema where any temporary tables are created as part of the execution of the data preparation tasks. By default, the schema that owns the selected driving table is used for hosting these tables, but it is advisable to create a dedicated database schema for exclusive use by Related Extract.

The schema selected should give the user running the extract permissions to create and drop tables, and it should provide sufficient space for the extract process. Depending on the amount of data to be extracted, the required space may be substantial. Compuware recommends allocating space that is at least twice the total size of all tables participating in the extract, or multiple extracts that may be simultaneously executed against same database.

## Tune-up Tips and Hints

The most critical part of performance tune-up for the Database Server-Driven Engine is the accurate setting of the data preparation threads and data processing threads options. This section describes how to initially estimate these values and how to adjust them in case of poor performance.

Typically, a good estimate for the data preparation threads value is the number of CPUs/cores on the database server multiplied by 2.5. Therefore, for a single quad-core processor, set this value to 10; for two quad-core processors, increase this value to 20, and so on. Usually this achieves the best possible performance for the data preparation tasks when the following conditions are satisfied:

- The database server is fully available for exclusive use by Related Extract for the total extract duration.
- The DBMS the extract is running against is not sharing access with other applications running on the same hardware and may be using significant processing resources. This includes other database servers running on the same system, any virtualization platforms, and so on.
- The extract specification includes a large number of tables and the vast majority of these tables are not extremely large. Typically, tables are considered extremely large when they contain tens or hundreds of millions of rows.
- The database server has an appropriate amount of available memory. The size of this appropriate amount of memory depends on the actual data, availability of indexes, DBMS, and other factors. Generally, a simple rule of thumb is to have about 1.5 GB to 2 GB of RAM per CPU available.

When one or more of these conditions are not satisfied, the ideal number of data preparation threads should be reduced accordingly. Some signs that this number should be reduced include:

- The **Running total** value shown in the **File-AID Processing Status** window does not vary for a long time.
- All CPUs on the database server are close to 100% utilization.
- Some OS- or database-specific performance counters show significant congestion.

A good first estimate for the data processing threads value is the number of CPUs/cores on the execution server, also multiplied by 2.5. This number also assumes similar conditions:

- The execution server is fully available for exclusive use by a single run of the Related Extract for the total extract duration.

- The execution server is not sharing access with other applications running on the same hardware and may be taking significant processing resources. This includes other database/application servers running on the same system, any virtualization platforms, and so on.
- Scalable I/O for the disk drive used for writing the extracted data. (Use disk performance monitoring tools to ensure it does not become a bottleneck for the selected number of threads.)
- No or limited impact of data privacy processing. (Some data privacy rules require that significant encryption or translation work be provided for each record/field causing a larger CPU consumption per thread.)

When one or more of these conditions are not satisfied, this number of data processing threads may need to be reduced. Some signs that this number needs to be reduced include:

- All CPUs on the execution server are about 100% utilized.
- The disk I/O counters point that writing to the disk may be a bottleneck.

Usually, if the same extract specification is expected to be executed repeatedly, it is worthwhile to execute several runs while adjusting the numbers of the data preparation and data processing threads, and then specify these numbers in the custom values on the **Advanced Settings** tab.

## Additional Notes

1. The **Advanced Settings** tab of the **Execution Options** dialog box is available only for SQL Server and Oracle database extracts. It is not shown for other databases, such as DB2 or Sybase, in which case the settings for the Execution Server-Driven Engine are used.

   When Related Extract for SQL Server or Oracle is run against the Execution Server-Driven Engine, the number of data preparation threads and the schema for temporary tables are ignored, but the setting for data processing threads replaces the **Extract_Max_Threads** value defined in the engine.properties file. In Release 5.3, this file is located by default at \ProgramData\Compuware\FAEX\Cfg (Windows Vista and later) or at \Documents and Settings\All Users\Compuware\FAEX\Cfg (prior to Windows Vista). In Release 5.2, this file is located by default at <*install folder*>/cfg. With the default value for this option set to **Server Setting**, it will use the same value as was used with the Execution Server-Driven Engine. However, if the user changes it, the new value will be honored by the execution server.

2. The Database Server-Driven Engine provides a much higher degree of parallel processing than the Execution Server-Driven Engine. While this provides significant performance benefits, it also increases the chance of exceeding the scalability limits of the database server or File-AID execution server hardware. For example, this brings the risk that other clients accessing the same database server would have slow response time or get no response at all for the duration of the extract.

   If this possibility is a significant concern, a recommended solution is to reduce the corresponding number of threads (data preparation threads) for the database server to a lower value. This will cause the extract to take a longer time to run but will keep the resource consumption to a lower level.

3. Another possible side effect from the higher degree of parallel processing is the consistency of results. While the results of the extract are expected to be the same for each run, in some rare situations, parallelization may cause some inconsistencies. This can happen only under the following two conditions:

   – There are circular relationships in the extract specification, meaning that some chains of the relationships make loops.

   – The extract scope excludes the sibling relationships and/or indirect dependent relationships.

The reason for the possible inconsistency is that, in the case of circular relationships, there is a possibility that the same data entries would be discovered from different directions (using different relationships), and depending on that their related records could be differently identified as siblings or indirect dependent, and therefore included or excluded from the extract.

This is a very rare situation that can also happen with the Execution Server-Driven Engine when more than one thread is set for the execution server. With the Database Server-Driven Engine, there is a higher probability for it to occur due to higher parallelization.

In any case, the extracted data is guaranteed to have referential integrity. If a possibility of inconsistent results exists and is deemed to be an issue, the available workarounds are to set the data preparation threads number to 1 or to include both sibling and indirect dependent relationships.

4. While the Database Server-Driven Engine is running, it creates tables in the database to temporarily keep data prepared for extraction. All of these tables are expected to be deleted automatically at the end of the extract process.

   However, in rare cases, some of these tables might not be deleted. For example, it can happen when the execution server loses connectivity with the database or in some cases of severe errors. When this happens, these tables can be deleted by using the File-AID DB Cleanup Utility or by any other means. For instructions on use of the DB Cleanup Utility, see the File-AID/EX Homebase online help.

   These tables are easy to locate in the database by the common prefix "FACS_RE_". They are created in the schema specified in the **Advanced Settings** dialog box or, by default, in the schema that contains the driving table for the extract. When deleting any of these tables, first ensure that no Related Extract job is running concurrently with the deletion.

# Performance Tuning for Release 5.1 and Earlier

## Related Extract Tuning Parameters

### BunchQuantity

The <*install folder*>/cfg/engine.properties file contains a line similar to:

```
BunchQuantity = { 1, 50, 45, 20, 5, 1 }
```

During a related extract, the execution server sends key data from one table back to the database in order to request related rows in other tables. For efficiency's sake, these keys are sent in "bunches" of more than one key; however, the optimal size of each "bunch" can vary depending on the number of columns that make up a key.

Therefore, BunchQuantity is set based on the number of columns per key as shown in Table 1-2:

**Table 1-2.** BunchQuantity Parameter Values

| BunchQuantity Parameter # | Number of Columns Per Key | Default Value | Description |
|---|---|---|---|
| 1 | N/A | 1 | Reserved for future use |
| 2 | 1 | 100 | The number of keys composed of 1 columns sent to the database at one time |
| 3 | 2 | 50 | The number of keys composed of 2 columns sent to the database at one time |

**Table 1-2.** BunchQuantity Parameter Values

| BunchQuantity Parameter # | Number of Columns Per Key | Default Value | Description |
|---|---|---|---|
| 4 | 3 | 33 | The number of keys composed of 3 columns sent to the database at one time |
| 5 | 4 | 25 | The number of keys composed of 4 columns sent to the database at one time |
| 6 | 5 | 1 | The number of keys composed of 5 or more columns sent to the database at one time |

For instance, if the current table that is being extracted uses a primary key that is composed of two columns, by default the bunch size is 50. In order to arrive at this value, look up 2 in the table in the 'Number of Columns Per Key' column and find the 'Default Value' in the same row. To change this setting, use the 'Bunch Quantity Parameter #' to know which parameter in the set to modify. In this case, it is the third parameter.

Another way to simplify setting the values is to just ignore the first parameter and remember that the second parameter is the number of keys for a 1 column key, the next for a 2 column key, then a 3 column key, and so on.

The bunch quantity parameters can be customized to optimize the speed of Related Extract. The default settings have been found to be good for the general case, but improvements may come from testing at both higher and lower settings for each.

## Extract_Max_Threads

The <*install folder*>/cfg/engine.properties file contains a setting called Extract_Max_Threads, which defaults to 5. This setting determines the maximum number of tables that a related extract will attempt to process at the same time. Due to the complexity of related tables, it is not always possible to process the maximum number of tables at once, such as when reading the first rows from the driving table.

In general Extract_Max_Threads should be set to between 1 and 1.5 times the number of CPUs on the execution server machine. The optimal value varies between Related Extract specifications and hardware configurations and should be arrived at experimentally.

There are cases where the execution server is located on a machine along with other large applications that need to share resources. In these cases it may be necessary to reduce the max threads to lower than the number of CPUs on the machine. Performance will likely slow down, but the machine can in this way be shared.

## CacheLookupTable

The <*install folder*>/cfg/engine.properties file contains a property called CacheLookupTable, which defaults to "true" but can be set to "false". When set to true, a related extract will cache entire lookup tables in memory for quick access during an extract. The amount of RAM taken up is described in "Data Disguise" on page 18.

Because of the time it takes to cache a lookup table, it is recommended that the cache not exceed 1.5 million rows, and preferably they can be significantly smaller. Although this introduces the chance of looking up duplicate values, that is common in undisguised data. For instance, if lookup tables are used to disguise first and last names, duplicate names are as likely to exist in the original undisguised data as the resulting disguised data. As long as the percentage of duplicate values is similar it does no harm in these cases.

## MappedMemory_Maximum_Threshold

The <*install folder*>/cfg/engine.properties file contains an optional property called MappedMemory_Maximum_Threshold. This setting is in megabytes and puts a ceiling on

the amount of memory that a Related Extract is allowed to map at any given time. When this threshold is reached, the job is stopped with an appropriate error message.

What can happen is that a large related extract can grow to consume more RAM than is available on the host machine, which causes it to start thrashing: swapping memory to and from disk without making substantial progress. As a protective measure, this property was added to artificially limit the amount of RAM that is to be used to keep the host machine from getting bogged down with swapping and to allow the machine to be shared by other users.

**Note:** This threshold must always be specified in megabytes, so a value of "12000" means 12 Gigabytes is the maximum amount of RAM that is allowed to be mapped at any one time.

***Example:***

Suppose the execution server is installed on a machine with 16 gigabytes of RAM that is shared by other users. Two gigabytes are to be reserved for the operation system and another four are reserved for other users. The correct setting is therefore:

```
MappedMemory_Maximum_Threshold = 10000
```

Or 10 gigabytes are reserved for the execution server.

When this threshold is reached, the remedies to the problem generally are these:

1. Add more RAM to the machine and increase the threshold.
2. Reduce the Extract_Max_Threads value to the point that this threshold will no longer be reached. See "Extract_Max_Threads" on page 14 and "Chase Keys" on page 15 for more details on using this property.
3. Do not run as many simultaneous related extracts as the threshold is the sum of the memory used by all running jobs.

Once one or more of these remedies has been applied, it will be necessary to rerun the extract specification from the beginning.

# Execution Server CPU Usage

Related Extract's CPU usage is governed by the Extract_Max_Threads tuning parameter detailed in "Extract_Max_Threads" on page 14.

In general a related extract will use just one CPU to start reading from the driving table, then one CPU per table related to the driving table, and so on through the chain of related tables. Up to Extract_Max_Threads CPUs will be used at once, but at some points fewer CPUs will be used due to natural bottlenecks in the relationships.

There is also a natural performance relationship between the database server machine and the execution server machine. One of the two is likely to be the performance bottleneck due to differing workloads. Adding CPUs to an execution server that has all of its CPUs busy most of the time should increase extract performance, but when the CPUs are mostly idle it will not provide an improvement. Similarly, increasing relevant resources on a database server that is the performance bottleneck will improve performance.

# Execution Server RAM Usage

## Chase Keys

Related Extract uses two types of RAM: heap memory and mapped memory. The heap memory does not need to be higher than 512 MB in most cases, but can be set higher

(using -Xmx?m) if needed. The mapped memory is what grows very large in a large extract.

Mapped memory contains two kinds of data per table being extracted:

1. **Key data:** The data from the table's key columns (unique and foreign).
2. **Index data:** An index that keeps track of which rows have already been extracted from the table.

The combination of the key data and index data are known as *chase keys*, which refers to the fact that Related Extract starts at the driving table and "chases" related rows from other tables in order to extract a relationally intact set.

There is a setting in <*install_dir*>/cfg/engine.properties called Extract_Max_Threads that determines the maximum number of tables an extract will attempt to process at the same time. In order to conserve memory, Related Extract only keeps chase keys in memory for the tables it is currently processing. Therefore, it is not necessary to have enough RAM to hold all of the chase keys in memory at once; only enough for Extract_Max_Threads tables. And since non-key columns are written to disk, they do not take up RAM.

## Memory Calculations

In order to calculate the amount of RAM required per table in the extract, add the two calculations together:

1. **Key data:** Add up the average number of bytes taken up by all key columns (unique and foreign) in a table and add about four bytes per column of overhead.

   There is one key data file for each foreign or unique/primary key in a table. Each key data file can have a maximum size of 1 Terabyte.

2. **Index data:** use the table below:

**Table 1-3.** Calculate RAM for Index Data

| Approximate # of Rows | RAM for Indices |
|---|---|
| < 328,000 | 4 megabytes |
| < 656,000 | 8 megabytes |
| < 1.32 million | 16 megabytes |
| < 2.625 million | 32 megabytes |
| < 5.25 million | 64 megabytes |
| < 10.5 million | 128 megabytes |
| < 21 million | 256 megabytes |
| < 42 million | 512 megabytes |
| < 84 million | 1 gigabyte |
| <168 million | 2 gigabytes |
| < 336 million | 4 gigabytes |
| < 672 million | 8 gigabytes |

**Note:** More than 200 million rows in a single table have not currently been tested. Larger numbers have been extrapolated for convenience.

## Other RAM Requirements

It should also be noted that the operating system and other users on a server will also require RAM. Therefore, it is often wise to leave 1-2 gigabytes of RAM just for the OS, and then an amount proportional to the requirements of other users. A system administrator

can help determine the appropriate RAM requirements of the system apart from Related Extract.

See "MappedMemory_Maximum_Threshold" on page 14 for details on how to place a ceiling on the amount of memory a Related Extract is allowed to map at any one time.

## Example

As an example, a schema was created that has a driving table with one million rows, and it has 20 related child tables that each have 200 million rows. Since selection criteria was not applied, the total row count to be extracted is therefore 4,001,000,000 (~4 billion).

For the key data per table: each 200 million row child table has only one key and each has a size of 5 bytes. Since the driving table has so many fewer rows than its child tables, its memory usage has been ignored in the following calculation.

***Key Data Calculation:***

200,000,000 rows * (5 bytes + 4 bytes overhead) == 1.8 gigabytes of key data per table

***Index Data Calculation:***

Lookup 200,000,000 rows in the above table == 4 gigabytes per table of index data

Therefore, how much RAM is required to process all 4 billion rows?

**Table 1-4.** Amount of RAM Required to Process Number of Simultaneous Tables

| Number of Simultaneous Tables Processed | RAM Required |
|---|---|
| 1 | 5.8 gigabytes |
| 2 | 11.6 gigabytes |
| 3 | 17.4 gigabytes |
| 4 | 23.2 gigabytes |
| 5 | 29 gigabytes |
| ... | ... |
| 20 | 116 gigabytes |

This particular test was run on a machine with 16 gigabytes of RAM, and using the table above it is easy to see that only two tables can be processed at the same time. Therefore, we must set Extract_Max_Threads=2 in order to extract the 4 billion row example. This will result in 2 of the 200 Million row tables being processed at a time.

A good rule of thumb is to take the number of rows of a few of the largest tables, extrapolate the average key size, then calculate and sum the key data + index data as shown above.

## Relational Distance

Any two tables in a related set of tables have a relational distance from each other, which is the number of "hops" or the number of tables that must be traversed before reaching each other. Because of this fact and because Related Extract only processes up to Extract_Max_Threads tables at the same time, two relationally distant tables may never end up being processed at the same time.

Therefore, in such cases, it may not be necessary to have enough RAM to process all large tables in a schema simultaneously. It is only necessary to have enough RAM for those tables that will be processed at the same time. However, because of the complexity of related data it can be difficult to identify ahead of time how relationally distant two tables must be before they are not processed at the same time.

### Data Disguise

Disguising data using any of the translation methods also adds RAM usage to a related extract when lookup table caching is turned on. See “CacheLookupTable” on page 14 for a description of how to set this variable.

When a lookup table is cached, it uses RAM in a very similar fashion to chase keys. The only difference is that the key data includes not only any lookup key column data but the return field data as well.

For instance, if “Translate - Search” is used with a lookup key that is an integer, and the returned field is also an integer, both of these values are stored inside the key data. So, from the example above, the impact of caching one lookup table with two 4-byte integers as lookup/return field data in a 1 million row lookup table would be as follows:

***Key Data Calculation:***

1,000,000 rows * [(4 bytes + 4 bytes overhead) * 2] == 16 megabytes of lookup/return data

***Index Data Calculation:***

Lookup 1,000,000 rows in the above table == 16 megabytes of index data

Therefore, caching this lookup table will require a minor 32 megabytes of RAM. Note that increasing the size to 20 million rows for the lookup table would require 576 megabytes RAM. For this reason, and because of the time it takes to cache a large lookup table, it is recommended that they not exceed 1.5 million rows.

## Execution Server Disk Usage

### Space Required

Running a related extract in an execution server uses disk in two ways: for the extracted data and for temporary files used to keep track of chase keys. Both are located in the Extract Data Location specified in the Related Extract “Prune relationships” screen under the Options dialog, File Locations tab.

The amount of space required for the temporary files is identical to the amount of RAM used by Related Extract as detailed in “Execution Server RAM Usage” on page 15.

However, the amount of disk space required for the extracted data can be difficult to quantify in advance. Although the amount of space required is proportional to the amount of data extracted, the exact amount varies depending on the size of the column types.

# Relationships and Subsetting Data

It is often the case that the amount of data stored in production tables is far greater than what is needed for test purposes. Because of this, it is often desirable to extract only a subset of the data since this will decrease the time to execute an extract, the size of the data stored on disk, and the load on the production server.

There are generally four features provided to adjust the amount of data extracted:

1. Selection Criteria provides the ability to specify exactly which rows are to be extracted from a table.
2. Excluding tables and/or relationships provides the ability to extract no rows from the affected tables.
3. Special handling for special tables. The process type options available for each table allow the user to specify how to handle tables with special considerations.

4. Global extract scope options.

## Selection Criteria

Selection Criteria provides a way to specify which rows are to be extracted from the table it is applied to, which reduces the number of rows extracted. Because of relationships between tables, this often causes a cascade effect that reduces the data extracted from other tables related to that table.

For example, assume that Table A is the driving table and it has selection criteria applied to it that causes only half of its rows to be extracted. If Table B is a related child of Table A, the only rows in B that will be extracted are those related to half of A's rows. Therefore, the selection criteria specified for Table A affects the number of rows extracted from Table B.

However, since there can be more than one table related to Table B, additional rows might be extracted from B through one of these other relationships. But since B is a child of A this might mean more rows need to be extracted from A to satisfy referential integrity or application relationships.

Another way to state this would be to say that otherwise there would be rows with foreign keys in Table B whose corresponding primary key in Table A would be missing from the extract because of selection criteria on Table A. This can cause those rows to be discarded by Related Loader.

Whether to satisfy RI/AR is left up to the user: on the “Prune relationships and set Selection Criteria” page of the extract wizard, users can right-click on a table that has selection criteria applied and check “Satisfy RI/AR”. This will cause additional rows that are not part of the selection criteria to be extracted from that table when it is needed to satisfy RI or AR. When this option is not checked, only rows that are part of the selection criteria will be extracted.

The *File-AID/EX Getting Started* guide describes how to apply selection criteria in the section Related Extract and Related Loader Tutorials/Tutorial 1: Extract Data from a Production System/Create Selection Criteria.

## Excluding Tables and/or Relationships

Another way to reduce the amount of data extracted is to exclude tables whose data is not needed. On the “Prune relationships and set Selection Criteria” page of the extract wizard, users can right-click on a table and select “Exclude”, which excludes that table from being extracted. It is also possible to right-click on a relationship line connecting two tables and select “Exclude”.

**Note:** When a table is excluded from being extracted, this can make other tables related to it unreachable, which essentially excludes them as well. One common strategy in subsetting the data extracted is to identify groups of tables whose data is not needed and exclude all relationships that connect them to the rest of the related set of tables.

When excluding tables, it is often helpful to switch to the Table List View or Relationship List View on the “Prune relationships” page of the extract wizard.

## Special Handling for Special Tables: Process Type

Another strategy to reduce the amount of data extracted is to recognize that certain tables should be treated specially. For each table in the “Prune relationships” page of the extract wizard, users can right-click on the table and select various “Process Type” options.

The available process type options are:

1. **All Rows:** This causes all rows of that table to be extracted regardless of whether they are related to rows in other tables. This is the only process type for unrelated tables.

   This option is useful in cases where extracting from one particular table will cause many unnecessary rows to be extracted from other related tables.

   **Note:** When this option is set, the rows extracted from this table will not cause rows to be extracted from its related tables. That is, in fact, the purpose of this feature.

2. **Passthrough:** No rows are extracted from this table, but related rows in other tables are extracted normally.

   This option is often used when there is a very large table that never changes and all of its rows should be extracted. However, because of the time it takes to extract it, it makes more sense to copy the table to its destination just once. Then, for instance, a nightly or weekly extract can be run to refresh the other tables in the specification.

3. **Relationship:** This is the default option. Rows are extracted based on their relationships with other tables.

## Extract Scope

The extract scope options are provided through the “Options...” button on the “Prune Relationships and Set Selection Criteria” page of the extract wizard. Detailed help is provided through the Help button on the Extract Scope tab.

These options provide a way to increase or decrease the global scope of an extract. If every possible related row is desired, check all of the options available. If the fewest number of rows is desired, deselect all but the most necessary of these options.

# Chapter 2.
# File-AID/EX Related Extract/Related Loader Benchmarks

The purpose of the benchmarks in this chapter is to provide recommendations for improving File-AID/EX 5.0 Related Extract and Related Loader performance while using SQL Server.

## Benchmark Environment

- CPU: 64-bit Windows machine with i7 Intel chip (8 processors)
- RAM: 16 GB
- Database: SQL Server 2008
- Extract: 77 million rows among 27 tables using RI
- Memory Usage - SQL Server: 7,941,848
- Memory Usage - File-AID/EX Execution Server: 1,677,196

***Timings — SQL Server***

*Execution Server-Driven Extract Engine*
- Extract of all 77 million rows (No Privacy): 17min 55 sec
- Extract of all 77 million rows (Privacy of 1 field in all 27 tables and File Encryption on all tables): 5hrs 57 min 56 sec

*Database-Driven Extract Engine*
- Extract of all 77 million rows (No Privacy): 06 min 09 sec
- Extract of all 77 million rows (Privacy of 1 field in all 27 tables and File Encryption on all tables): 3 hrs 24 min 59 sec

*Related Loader*
- Load of all 77 million rows (default of commit every 100 rows): 5 hrs 35 min 36 sec

## Recommendations

These recommendations will aid users in fine tuning their systems to achieve better performance. The sections that follow give detailed instructions on how to modify system settings based on these recommendations.

- Reduce workload contention on database server.
- Create pagefile on dedicated drive.
- Configure initial pagefile size equal to 1.5 times installed RAM.
- Monitor hard page faults. Should average less than 10 pages per second.
- Ensure I/O data transfer time averages less than 20 milliseconds.
- Monitor NIC transfer rate.
- Manage transaction logs in SQL Server references.

- Multiple processors
- Memory Requirements: 9.5 GB
- Increase Related Loader Commit Every nth row

## Reduce Workload Contention on Database Server

All processes running on the same server compete for finite system resources: CPU, memory, and I/O. It is important to monitor and eliminate unnecessary processes that may place excessive load on a server such that the server performance is degraded, or such that the server itself is so heavily loaded that it fails to perform routine housekeeping for its own maintenance.

There are three tools that can be used to monitor and manage processes running on the server:

**MSCONFIG** — Usually included with Windows. Use to control which processes run at startup.

**Task Manager** — Included with Windows. Use to monitor and manage processes after system startup.

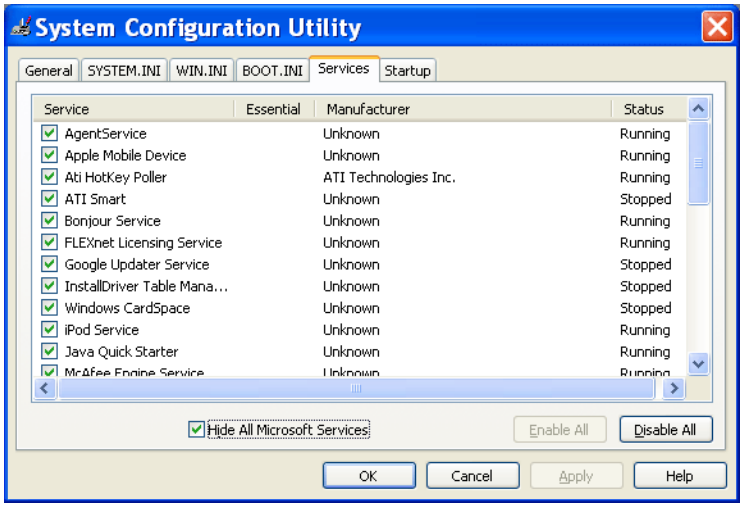
**Process Explorer** — A free download from Microsoft. Like Task Manager, use it to monitor and manage processes after system startup. Process Explorer can be configured to replace Task Manager. It provides more information such as process author, registry keys, and DLLs used.

### MSCONFIG

To use MSCONFIG to stop a service or application:

1. From the **Run** dialog box, type **msconfig** and click **OK**. The **System Configuration Utility** dialog box appears.

**Figure 2-1.** System Configuration Utility — Services Tab




2. Click the **Services** tab.
3. Select **Hide All Microsoft Services**.
4. Deselect any services that are not needed.

5. Click the **Startup** tab and deselect (to deactivate) any unnecessary applications that were activated during startup.
6. Click **OK**.
7. Restart the computer for the changes to take effect.

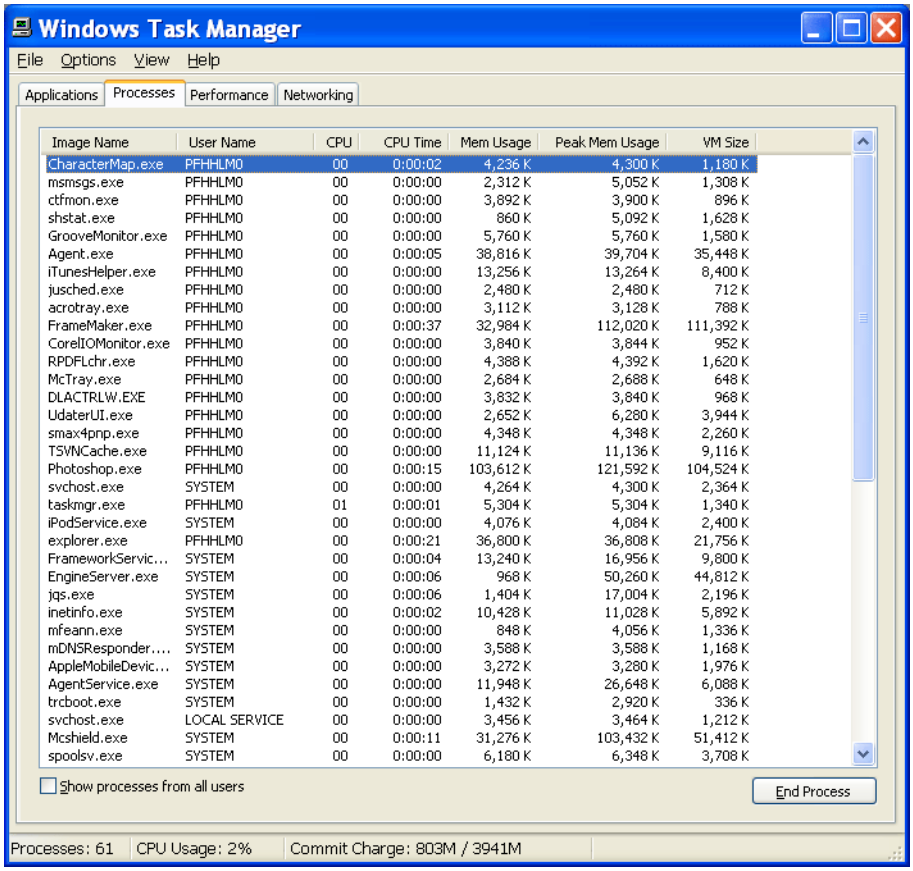
## Task Manager

Task Manager allows users to monitor and stop processes, although it may be difficult to identify the vendor or author of the process. Process Explorer may be preferable instead.

To use Task Manager to stop a process:

1. Right-click the Windows Taskbar and select **Task Manager**. The **Windows Task Manager** dialog box appears.
2. Click the **Processes** tab.

**Figure 2-2.** Windows Task Manager




3. From the list, select the process to be stopped and click **End Process**.
4. Close the Task Manager.

## Process Explorer

To obtain detailed information about a questionable process, Process Explorer is a good tool to use. In addition to resource usage, Process Explorer provides data on DLLs, registry keys, vendor and author, and process dependency information.

Although Process Explorer is not included with Windows by default, it can be downloaded freely from Microsoft. The download is a ZIP file that can be extracted into any folder. Once extracted, simply double-click procexp.exe. For convenience, users may want to create a desktop shortcut or replace Task Manager with Process Explorer.

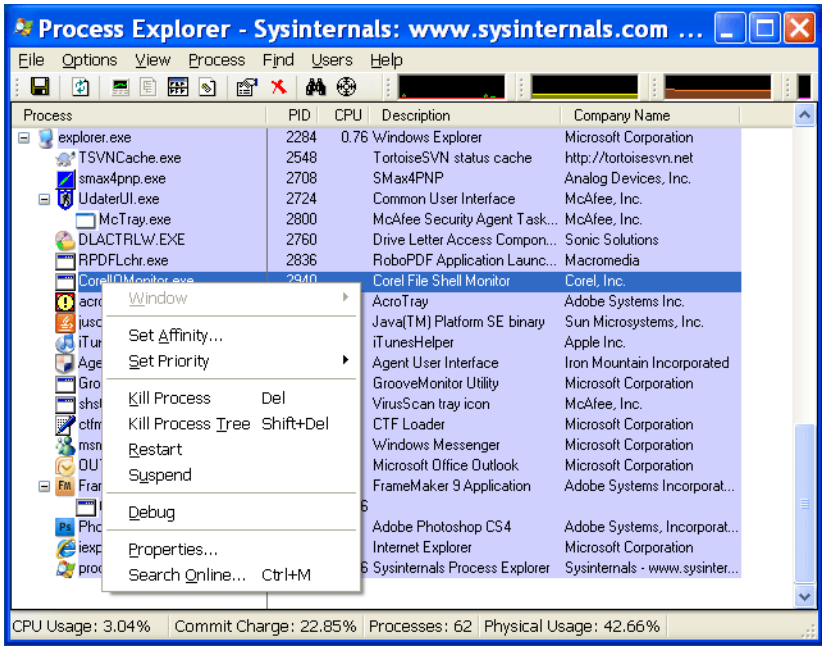
Once started, Process Explorer displays a hierarchal view of loaded processes. Processes that were started by another process are indented.

The 'Company Name' column facilitates the process' vendor and helps users determine which process to stop.

To stop a process:

1. Start Process Explorer. The **Processor Explorer** dialog box appears.

**Figure 2-3.** Process Explorer



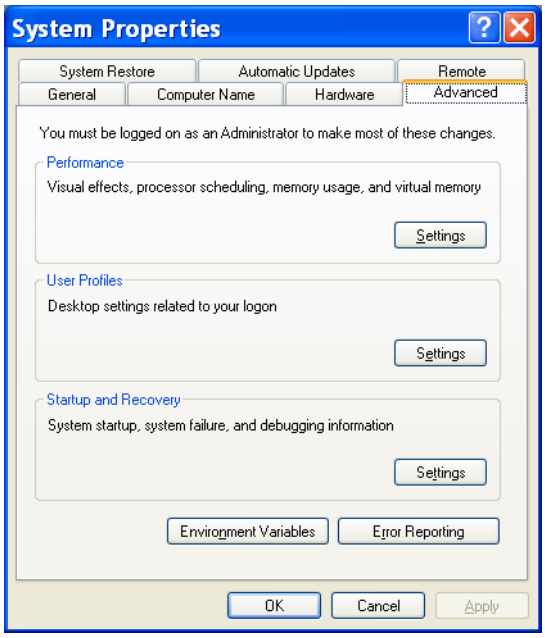

2. Right click a process in the list and select **Kill Process**.
3. Close Process Explorer.

## Configure Initial Pagefile Size Equal to 1.5 Times Installed RAM

To change your initial pagefile size:

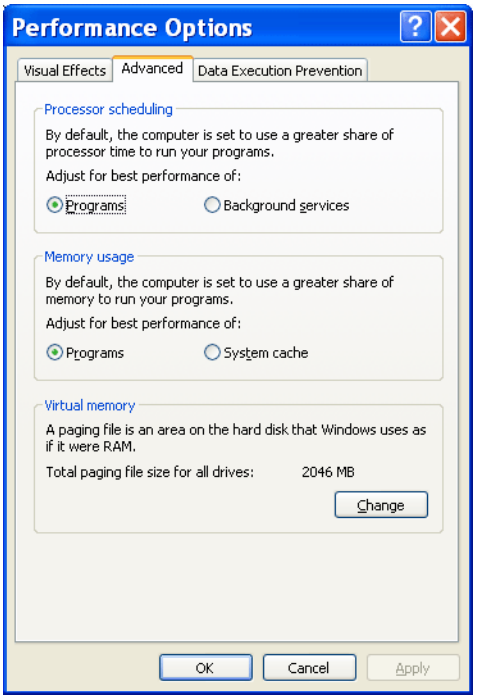
1. From the **Control Panel**, double-click **System**. The **System Properties** dialog box appears.

**Figure 2-4.** System Properties — Advanced Tab




2. Click the **Advanced** tab.
3. In the **Performance** box, click **Settings**. The **Performance Options** dialog box appears.

**Figure 2-5.** Performance Options — Advanced Tab

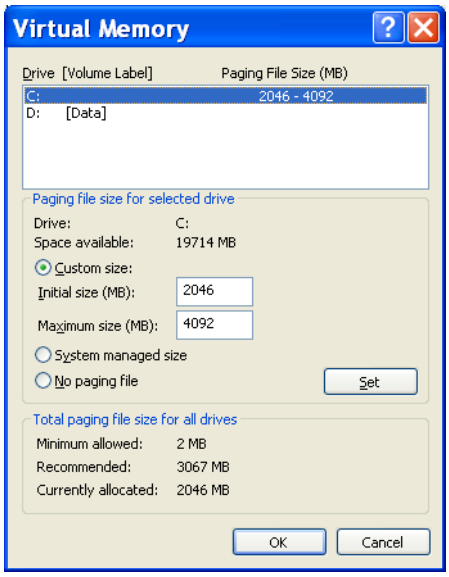



4. Click the **Advanced** tab.
5. In the **Virtual memory** box, check that the **Total paging file size for all drives** amount listed is at least 1.5 times the amount of installed RAM.

If this is not the case, continue with the next step; otherwise no changes are necessary.

6. Click **Change** to adjust the paging file size. The **Virtual Memory** dialog box appears.

**Figure 2-6.** Virtual Memory




7. Select the appropriate drive.
8. Edit the **Initial size** and **Maximum size** values to increase the amount of virtual memory.
9. Click **Set** to finalize the change.
10. Click **OK** three times and close the **Control Panel** dialog box.

# Using Performance Counters

Page faults, data transfer time, NIC transfer rate, as well as many other performance metrics can be monitored using Microsoft's Performance Counters.

Depending on needs, performance counters can be displayed in two ways:

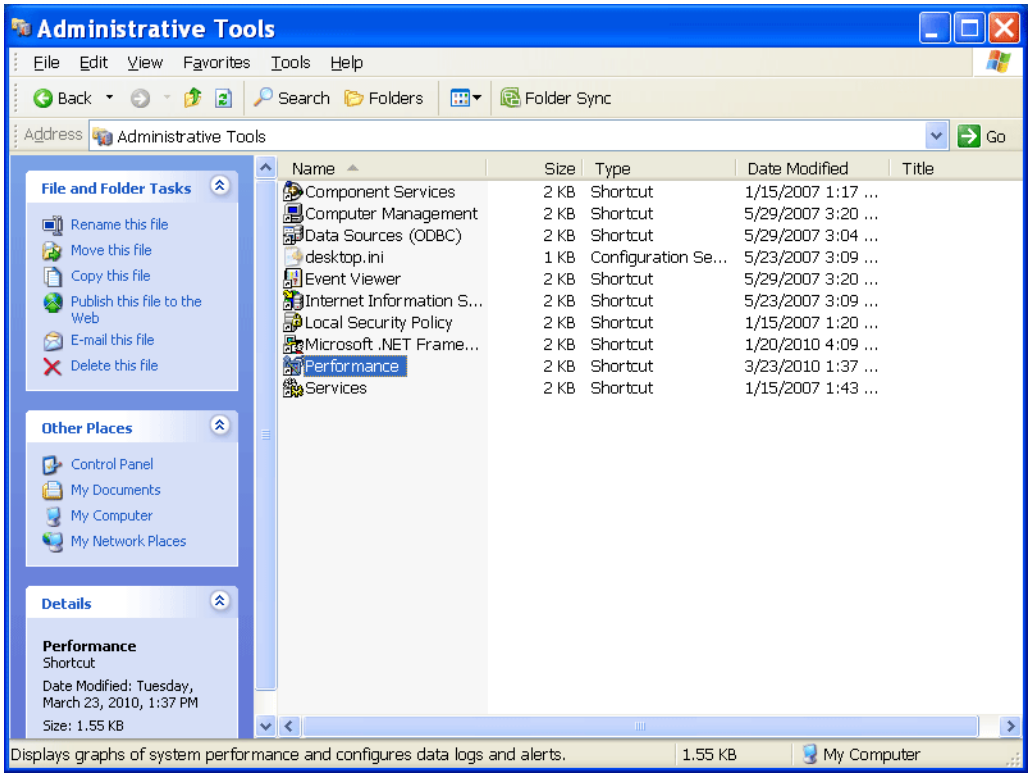
1. **System Monitor** for real-time analysis — values are graphically displayed immediately.
2. **Performance Log** for historical analysis — values are captured in a file for future study.

## Use the System Monitor Real-time Analysis

To monitor a system in real-time:

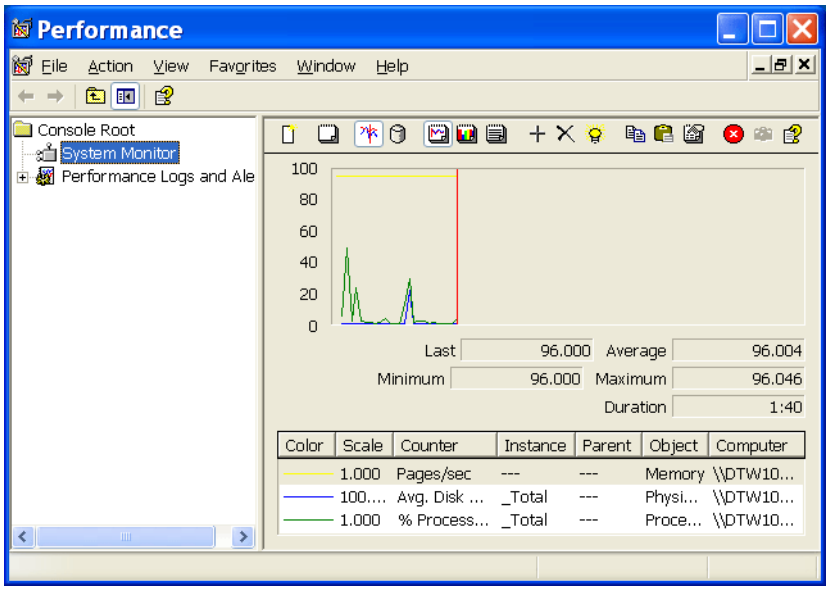
1. From the **Control Panel,** double-click **Administrative Tools**. The **Administrative Tools** dialog box appears.

**Figure 2-7.** Administrative Tools



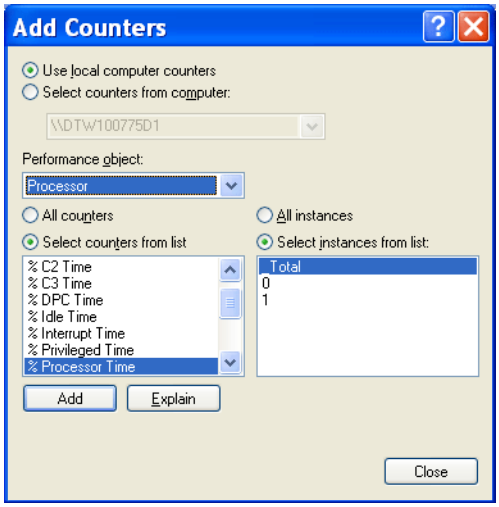

2. Double-click **Performance**. The **Performance** window appears.

**Figure 2-8.** Performance Window




3. Select **System Monitor** and click the **plus (+) sign** on the monitor toolbar. The **Add Counters** dialog box appears.

**Figure 2-9.** Add Counter Dialog Box



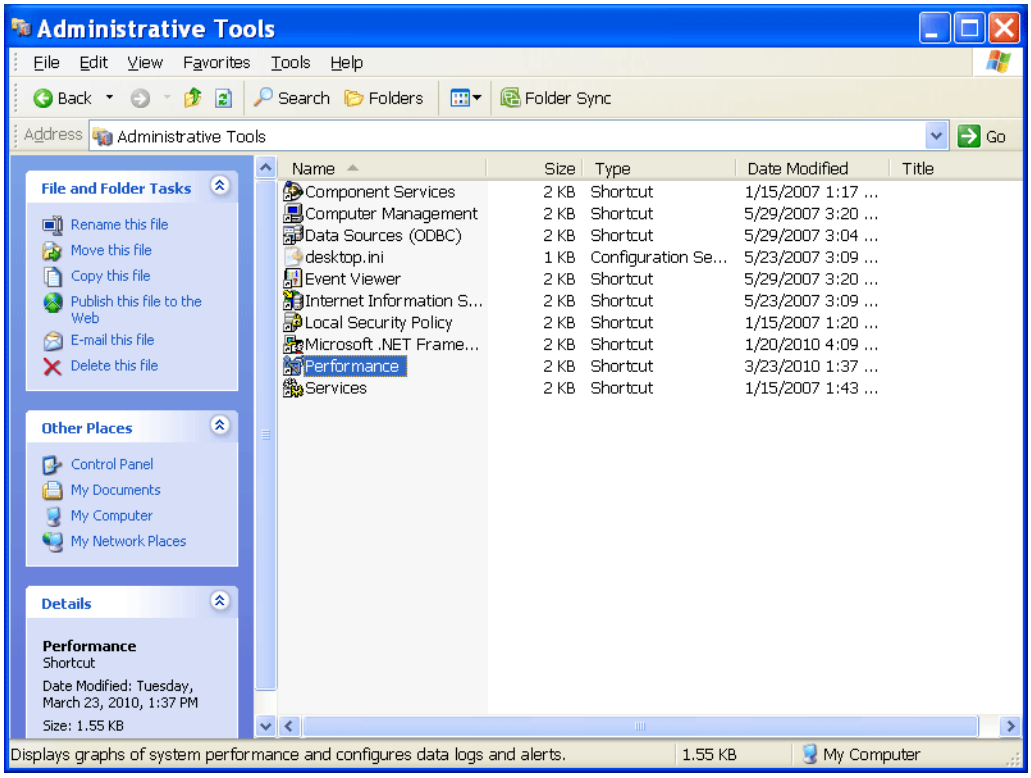

4. Select **Use local computer counters**, select a **Performance object** from the list, and select a **Counter** from the counter list.
5. Click **Add** to add the counter to the log file.
6. Click **Close**.

## Create a Performance Log File

To create a performance log:

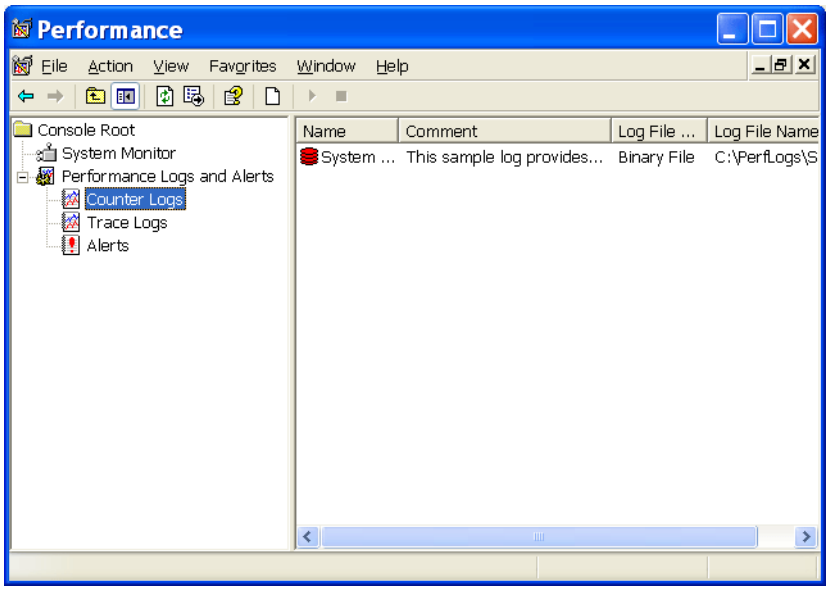
1. From the **Start** menu, select **Control Panel**.
2. Click **Administrative Tools**. The **Administrative Tools** dialog box appears.

**Figure 2-10.** Administrative Tools



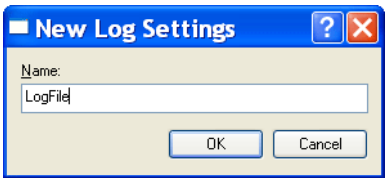

3. Click **Performance**. The **Performance** dialog box appears.

**Figure 2-11.** Performance Window — Counter Logs



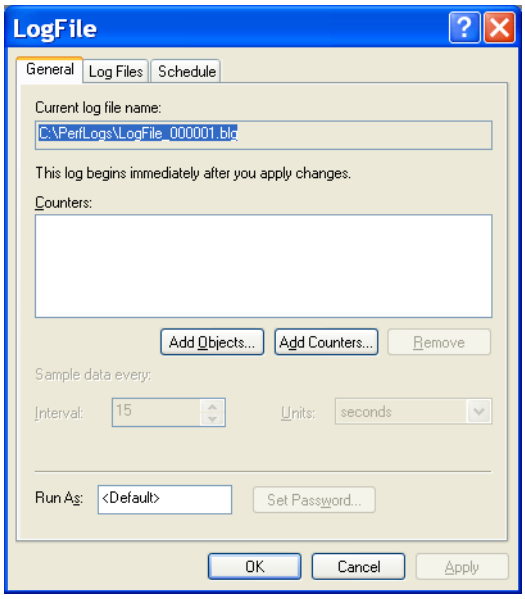

4. Expand **Performance Logs and Alerts** and select **Counter logs**.
5. Right-click **Counter logs** and select **New Log Settings**. The **New Log Settings** dialog box appears.

**Figure 2-12.** New Log Settings Dialog Box



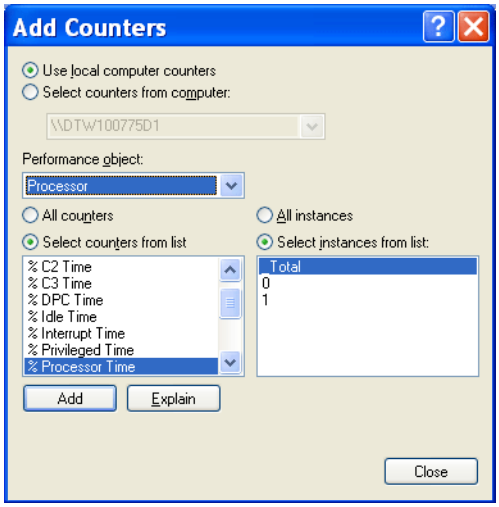

6. Type a file name for the counter log and click **OK**. A dialog box appears titled with the name you specified on the **New Log Settings** dialog box. In this example, we named the file **LogFile** so the window is named LogFile.

**Figure 2-13.** LogFile Dialog Box



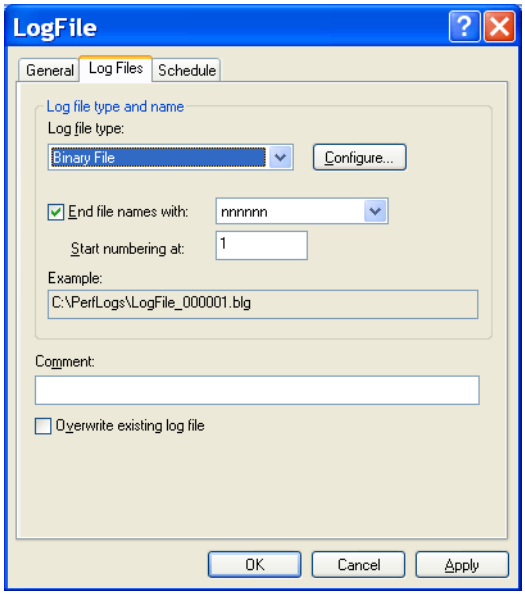

7. Click **Add Counters**. The **Add Counters** dialog box appears.

**Figure 2-14.** Add Counter Dialog Box



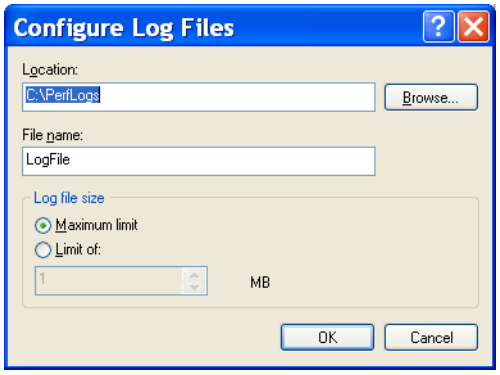
8. Select **Use local computer counters**.
9. Select a performance object from the **Performance object** list and a counter from the counter list and click **Add**.
10. Click **Close**. Focus returns to the previous window (in this example, the LogFile dialog box).
11. Select the **Log Files** tab.

**Figure 2-15.** LogFile Window - Log Files Tab



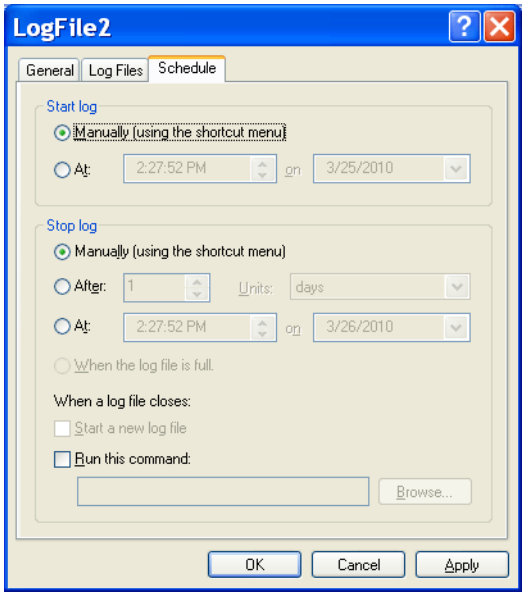

12. Select the **Log file type** from the list and click **Configure**. The **Configure Log Files** dialog box appears.

**Figure 2-16.** Configure Log Files




13. In the **Location** field, type or browse to the location in which to store the log file. The file name is prefilled with the name specified earlier.
14. Under **Log file size**, specify the size for the log file.
15. Click **OK**. The LogFile dialog box appears.
16. Select the **Schedule** tab.

**Figure 2-17.** LogFile — Schedule Tab




17. Choose **Manually** for starting and stopping the log and click **OK**. The **Performance** dialog box appears.
18. Right-click on the newly created log file and select **Start** to activate logging.
19. To stop the log file, right-click on the file and select **Stop**.

The log file can now be imported into Microsoft Excel or processed by a user-created program.

## Monitor Hard Page Faults

When monitoring hard page faults, the average should be less than 10 input pages per second.

Input pages per second is an indication of system memory contention. It is commonly called hard page faults because the memory manager satisfies the page fault by accessing the pagefile (disk) to retrieve the needed page from disk to RAM. Disk I/O is much slower than RAM. Access times (the time to read or write data) on RAM are measured in nanoseconds (billionths of a second) and on disks in milliseconds. Excessive hard page faults have an extremely detrimental impact on application and system performance.

Use **Performance Counter Input pages/sec** of memory object to monitor hard page faults.

## Ensure I/O Data Transfer Time Averages Less than 20 Milliseconds

I/O data transfer time records the duration of an I/O operation to or from disk. Monitor this value over time. If average I/O transfer time exceeds 20 milliseconds on the database server, another process' file may be contending with and slowing down the database. Move the offending file to another drive.

I/O contention may also occur on the pagefile drive, which is slowing down paging operations. Again, isolate your pagefile on a dedicated device or poor performance may result.

Use Performance Counter: **Avg Disk sec/Transfer** of physical disk object.

## Monitor NIC Transfer Rate

Monitoring NIC transfer rates reports the rate at which the network card is sending and receiving network data. If this rate is significantly lower than expected given the speed of the network and network card, something may be wrong with the card. As a simple test, use the PING command.

Use Performance Counter: **Bytes Total/sec** of the network interface object.

## Manage Transaction Logs in SQL Server (References)

Some sites have difficulty controlling SQL Server Transaction Logs growth. Listed below are several references that describe the impact of log growth, the steps that should be taken to minimize that impact, and the causes of unexpected growth:

- INF: Causes of SQL transaction log filling up

  http://support.microsoft.com/default.aspx/kb/110139

- A transaction log grows unexpectedly or becomes full on a computer that is running SQL Server

  http://support.microsoft.com/default.aspx/kb/317375

- Shrinking the transaction log

  http://msdn.microsoft.com/en-us/library/aa174524.aspx

- How to stop the transaction log of a SQL Server database from growing unexpectedly

  http://support.microsoft.com/default.aspx/kb/873235

## Multiple Processors

Because Related Extract is a multi-threaded process, multiple tables can be extracted at the same time. In engine.properties, there is a variable, Extract_Max_Threads, that should be set to one less than the number of processors on the CPU.

## Memory Requirements

SQL Server uses 7,941,848 of memory and File-AID/EX Execution Server uses 1,677,196 of memory, totaling approximately 9.5 GB. With less memory than this, constant paging will occur, which will make the process take much longer to complete.

## Increase Related Loader Commit Every nth Row

The default commit rate on the Related Loader Load Options screen is every 100th row. This is appropriate if problems during Related Loader execution are expected. If, problems during execution are not expected, processing time can be significantly reduced by increasing this number, even as high as 10000.

# Compuware's Benchmark Lab Environment

## Hardware

**Table 2-1.** Benchmark Environment — Hardware

| | EXECUTION CLIENT | DATABASE SERVER |
|---|---|---|
| CPU | Intel Xeon x5355<br>8 cores @ 2.66 GHz | Intel Xeon x5355<br>8 cores @ 2.66 GHz |
| MEMORY MODEL | 64-Bit | 64-Bit |
| VIRTUAL STORAGE | RAM - 16 GB<br>Pagefile - 24 GB | RAM - 16 GB<br>Pagefile - 24 GB |
| HDD | C: 19.6 GB<br>D: 499.0 GB<br>S: 25.3 GB (Pagefile) | C: 20.5 GB<br>D: 1.31 TB<br>S: 30.2 GB (Pagefile) |

## Software

**Table 2-2.** Benchmark Environment — Software

| | EXECUTION CLIENT | DATABASE SERVER |
|---|---|---|
| OPERATING SYSTEM | Windows Server 2003<br>.Net 3.5 SP1 | Windows Server 2003<br>.Net 3.5 SP1 |
| JAVA | 1.7.0_14 | --- |
| FILE-AID/EX | Version 5.3 | --- |
| ORACLE | --- | 11G |
| SQL SERVER | Version 10.0.1600 | Version 10.0.1600 |

# Related Extract Runtime Database Tables

Following is the generic tables and number of rows extracted that were used in all benchmarks. See "Timings — SQL Server" on page 21 for length of time to run.

**Table 2-3.** Related Extract Runtime Database Tables

| Table name | Current Rows Extracted |
|---|---:|
| guest.CUSTOMERS | 8,705,307 |
| guest.CALL_TRACKING | 1,000,000 |
| guest.CUSTOMER_ENROLLMENTS_NEW | 903,000 |
| guest.CUSTOMER_DATA_NEW | 1,000,000 |
| guest.CUSTOMER_ENROLLMENTS_HST | 903,000 |
| guest.CUSTOMER_IDS_HST | 8,705,307 |
| guest.CUSTOMER_DATA_HST | 1,000,000 |
| guest.CUST_COV_DSNR_HST | 1,000,000 |
| guest.CUSTOMER_IDS_NEW | 8,705,307 |
| guest.CUSTOMER_PROVIDERS_HST | 11,041,542 |

**Table 2-3.** Related Extract Runtime Database Tables

| Table name | Current Rows Extracted |
|---|---|
| guest.CUSTOMER_PROVIDERS_NEW | 11,041,542 |
| guest.CUSTOMER_RELATIONS | 1,000,000 |
| guest.CUST_COV_DSNR_NEW | 1,000,000 |
| guest.HRA_HST | 1,000,000 |
| guest.HRA_NEW | 1,000,000 |
| guest.LETTERS | 1,000,000 |
| guest.MARKETING_ENROLLMENTS | 11,000,000 |
| guest.QUESTION_ANSWERED_HST | 1,000,000 |
| guest.QUESTION_ANSWERED_NEW | 1,000,000 |
| guest.VALIDATIONS | 1,000,000 |
| guest.CUST_ENR_RECON_HST | 2,308 |
| guest.CUST_ENR_RECON_NEW | 82,014 |
| guest.CUST_MEDICARE_DATA_HST | 1,000,217 |
| guest.CUST_MEDICARE_DATA_NEW | 632,691 |
| guest.CUSTOMER_COMMENTS | 1,000,000 |
| guest.CUSTOMER_COVERAGES_HST | 791,377 |
| guest.CUSTOMER_COVERAGES_NEW | 791,377 |
| **Running total** | **77,304,989** |

# Related Extract Relationship Criteria Benchmark

## SQL Server Database Tables

Extract Specification: Cpwr_Benchmark_No_Privacy
Database Server Type: Microsoft SQL Server
Database Connection: jdbc:jtds:sqlserver://dtw-facsdevtst:1433/pubs
Execution Server Name: LOCAL
Execution Server Location: LOCAL
Date: 12/19/2010
Time: 1:08 PM

**Table 2-4.** Related Extract Relationship Criteria - SQL Server

| Parent Table | Dependent Table | Type | Dependents | Status |
|---|---|---|---|---|
| guest.CUSTOMERS | guest.CALL_TRACKING | RI | * | |
| guest.CUSTOMERS | guest.CUST_COV_DSNR_HST | RI | * | |
| guest.CUSTOMERS | guest.CUST_COV_DSNR_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUST_ENR_RECON_HST | RI | * | |
| guest.CUSTOMERS | guest.CUST_ENR_RECON_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUST_MEDICARE_DATA_HST | RI | * | |
| guest.CUSTOMERS | guest.CUST_MEDICARE_DATA_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_COMMENTS | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_COVERAGES_HST | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_COVERAGES_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_DATA_HST | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_DATA_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_ENROLLMENTS_HST | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_ENROLLMENTS_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_IDS_HST | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_IDS_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_PROVIDERS_HST | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_PROVIDERS_NEW | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_RELATIONS | RI | * | |
| guest.CUSTOMERS | guest.CUSTOMER_RELATIONS | RI | * | |
| guest.CUSTOMERS | guest.HRA_HST | RI | * | |
| guest.CUSTOMERS | guest.HRA_NEW | RI | * | |
| guest.CUSTOMERS | guest.LETTERS | RI | * | |
| guest.CUSTOMERS | guest.MARKETING_ENROLLMENTS | RI | * | |
| guest.CUSTOMERS | guest.QUESTION_ANSWERED_HST | RI | * | |
| guest.CUSTOMERS | guest.QUESTION_ANSWERED_NEW | RI | * | |
| guest.CUSTOMERS | guest.VALIDATIONS | RI | * | |

# Related Loader Benchmark

## Load Summary

Extract Input: D:\root\ExtractedData\pfhcmb1\BENCHMARK_TEST
DataBase: dtw-facsdevtst.pubs
Execution Server Name: LOCAL
Execution Server Location: LOCAL

## Related Loader Inserts, Deletes, Replaces, and Discards

The following tables were created with the number of inserts specified. Refer to "Timings — SQL Server" on page 21 for length of time to run Related Loader using SQL Server.

**Table 2-5.** Related Loader Deletes, Inserts, Replaces, and Discards Benchmark

| Object Name | Deletes | Inserts | Replaces | Discards |
|---|---|---|---|---|
| guest.CUSTOMERS | 0 | 8705307 | 0 | 0 |
| guest.VALIDATIONS | 0 | 1000000 | 0 | 0 |
| guest.QUESTION_ANSWERED_NEW | 0 | 1000000 | 0 | 0 |
| guest.QUESTION_ANSWERED_HST | 0 | 1000000 | 0 | 0 |
| guest.MARKETING_ENROLLMENTS | 0 | 11000000 | 0 | 0 |
| guest.LETTERS | 0 | 1000000 | 0 | 0 |
| guest.HRA_NEW | 0 | 1000000 | 0 | 0 |
| guest.HRA_HST | 0 | 1000000 | 0 | 0 |
| guest.CUSTOMER_RELATIONS | 0 | 1000000 | 0 | 0 |
| guest.CUSTOMER_PROVIDERS_NEW | 0 | 11041542 | 0 | 0 |
| guest.CUSTOMER_PROVIDERS_HST | 0 | 11041542 | 0 | 0 |
| guest.CUSTOMER_IDS_NEW | 0 | 8705307 | 0 | 0 |
| guest.CUSTOMER_IDS_HST | 0 | 8705307 | 0 | 0 |
| guest.CUSTOMER_ENROLLMENTS_NEW | 0 | 903000 | 0 | 0 |
| guest.CUSTOMER_ENROLLMENTS_HST | 0 | 903000 | 0 | 0 |
| guest.CUSTOMER_DATA_NEW | 0 | 1000000 | 0 | 0 |
| guest.CUSTOMER_DATA_HST | 0 | 1000000 | 0 | 0 |
| guest.CUSTOMER_COVERAGES_NEW | 0 | 791377 | 0 | 0 |
| guest.CUSTOMER_COVERAGES_HST | 0 | 791377 | 0 | 0 |
| guest.CUSTOMER_COMMENTS | 0 | 1000000 | 0 | 0 |
| guest.CUST_MEDICARE_DATA_NEW | 0 | 632691 | 0 | 0 |
| guest.CUST_MEDICARE_DATA_HST | 0 | 1000217 | 0 | 0 |
| guest.CUST_ENR_RECON_NEW | 0 | 82014 | 0 | 0 |
| guest.CUST_ENR_RECON_HST | 0 | 2308 | 0 | 0 |
| guest.CUST_COV_DSNR_NEW | 0 | 1000000 | 0 | 0 |
| guest.CUST_COV_DSNR_HST | 0 | 1000000 | 0 | 0 |
| guest.CALL_TRACKING | 0 | 1000000 | 0 | 0 |

# Index

## B



## C



## D



## E



## H



## I



## K



## L



## M



## N



## P



## R

## S



## T



## V