HP Connect-It

For the Windows®, Linux®, and UNIX® operating systems

Software Version: 9.52

Improving Performance

Document Release Date: June 2013

Software Release Date: June 2013

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Contents

How to Use this Guide

Chapter Connect-It - Presentation
This chapter presents an overview of how Connect-It works.

Chapter Optimization - Source connector
This chapter presents the different Connect-It Source connector parameters that impact the performance of a scenario.

Chapter Optimization - Destination connector
This chapter presents the different Connect-It Destination connector parameters that impact the performance of a scenario.

Chapter How to improve the performance of a scenario - Example
This chapter explains how to modify a scenario in order to improve processing performance.

Chapter 1

Introduction to Improving Performance

This Guide describes the HP Connect-It software and how to improve its performance. Various performance improvement methods are described, including:

- Parameter Management
- Processing Data
- Reconciliation Cache Scenario Configuration
- Scenario Modification
- Parallelization
- Customizing the .cfg file
- Reconciliation Keys

Audience

This Guide is aimed at anyone who needs to perform an integration using Connect-It.

Processing Data

Connect-It interacts with external applications via connectors (also called components). During a data transfer from one database to another:

- A source connector produces documents.
  Each document corresponds to a data container that is found in the source application.
- A source connector produces documents. A mapping box reorganizes the structure of the documents produced by the source connector which enables them to be consumed by the destination connector.
- A destination connector consumes documents.
  Each document corresponds to a data container that is found in the source application.

When using database-type connectors, each document corresponds to a record in a table (and possibly to a link to other tables).

For other connectors, the data containers correspond to e-mail messages, delimited text files, security information, etc.

The connectors you choose, how they are related, and the definition of mappings between source and destination type documents are all used to create an integration scenario. After the testing and debugging phase, a scenario is associated with a schedule and then put into production.

As with all other integration processes, these documents can either be partially or totally rejected during the operation. By using processing reports and document logs, you can reprocess rejected documents without having to redefine your entire integration scenario.

Specific Terminology

Injector

An injector is any element that injects data into a connector. It can be a server or a connector.

Chapter 2

Optimization - Source connector

This chapter presents the different Connect-It parameters that impact the performance of a scenario.

Retrieving the Correct Element

Data format

Data formats must be consistent between the source database and the destination database.

If data formats are not the same between the source database and the destination database, then the data must be processed in the mapping or before being inserted into the destination database. Processing has a direct, negative impact on performance. It is preferable to perform processing at the database level for each database rather than having Connect-It handle the processing tasks.

If processing is not performed, records may be duplicated or superfluous records may be inserted into the database.

- A common example where data formatting is involved is for text type fields.
  For example, in the source database, last names are formatted as "Abcde" and in the destination database as "ABCDE". In this case, you could apply a processing rule in the source application to format text type elements to match the formatting in the destination database. By doing this, you do not need to perform additional queries in the mapping or during data reconciliation in Connect-It.

- Also, you need to check that strings are correctly processed and that no unnecessary blank spaces are appended to them. If blank spaces are added, then records may be duplicated during data reconciliation and reconciliation may not be performed on the requested fields.

Using a Where clause to filter relevant items

This section only concerns database type connectors.

Applying a Where clause reduces the number of documents that are produced by the source connector and thus improves processing times by analyzing only pertinent items.

When a user applies a filter on the document type to be produced (via a WHERE clause), the user must ensure that the fields which are targeted by the filter are indexed. Otherwise the query may be less efficient.

Optimizing the use of a WHERE clause

The Database-type connectors have production directives that enable you to enter WHERE clauses.

The query must be performed on indexed fields. Otherwise it will be performed on the entire table.

Pointers

Scheduling pointers operate in the following manner: Connect-It saves the status of the pointer and sets a boundary between what has been processed and what has not (oftentimes, the date-time field contains a date that corresponds to the last time a document or record was processed).

Pointers can be indexed in order to reduce the amount of data to process in a scenario.

In most cases, the status of the pointers concerns a specific date-time field. Information provided by the pointer is used to manage the processing of data from a third-party application following a specified user context.

Example: A source connector only processes the records from an Employees table corresponding to the newly hired employees by using each employee's hire date as the pointer.

The pointer type that is used depends on the third-party application and the produced document type.

Indexing pointers

If the field used as the pointer is not indexed, the entire table will be searched by the database engine in order to retrieve the relevant documents. Doing so will increase the time required to produce the document.

For example, it is more difficult to retrieve 10 documents from a table that contains 10 million non-indexed documents than a from table that has an index every 10,000 records.

In order to reduce time required to produce documents, fields that are used as pointers must be indexed.

For example, when the scheduling of a scenario is based on scheduling pointers, only records that have been modified since the last time the scenario was executed will be selected for production.

Select only columns that are pertinent to the scenario

For the document type produced by the connector, only select fields that are relevant to the data that is retrieved. For example, for a query on the departments and employees table, you may want to only retrieve an employee's identifier and none of the data from the other columns.

Improving the Production of Documents

Improving a connector's document production involves obtaining the data from a source application as rapidly as possible.

This section explains the different ways you can reduce the time it takes to produce documents.

Optimizing the server of the source data

The production of documents depends largely on the use of the source data's server, as well as its connection.

Connect-It does not perform numerous conversions or process large amounts of data when in production mode.

To improve the results, verify:

- The load of the source data server
  The load depends on the server's technical capacity: processor, available memory, etc.
- The network connection (WAN, LAN).

Network latency

Ideally, the applications will be located on the same server. If this is not the case, we recommend that Connect-It be installed in close proximity to the server with which Connect-It interacts the most.

Checking queries executed by the connector

Connect-It does not perform numerous conversions or process large amounts of data when in production mode.

To track the queries transmitted to the source database, select the Connector/ Show tracking lines option via the **Edit/ Options** menu.

After you configure this option, the Document log will display messages corresponding to the queries executed by your scenario's source connectors.

Example:

```
SELECT AcctCode,AssetTag,BarCode,dDispos,DisposProfit,dtListPriceCv,Field2,FullN
ame FROM amAsset
```

This information enables you to:

- Verify that the request is generated in compliance with the defined document type and the production directives that have been set.

- Check that the query syntax is relevant and that is concerns the requested object.
- Relaunch the query using a third-party tool and observe the time it takes to execute it.

Execute indexed queries to retrieve a data set

In most cases, to retrieve a data set, processing performance will be increased if the reconciliation keys that are used in the query (WHERE clause) correspond to one or more indexed fields.

Note
The use of indexes and the resulting increase in processing performance depend on the database engine and the connector being used.

In general, the use of an index can be beneficial if less than 5% of all source records are to be retrieved. On the contrary, the use of an index is not beneficial if almost all the data has to be processed.

Calculating the number of documents to produce

This option lets you count the number of documents that will be processed by the source connector.

This value enables you to display a progress bar in the Scenario builder. This bar will indicate the number of documents processed by the source connector.

For an Asset Manager or Database connector, this option triggers a SELECT COUNT type query.

```
SELECT COUNT(AcctCode) FROM amAsset
```

Example:

Warning

Generally, this option has a significant, negative impact on performance. We advise you to disable this option in production mode.

To disable the calculation of the number of documents that a source connector must produce:

1. Select **Edit/ Options**.
2. Unfold the Connector node.
3. Set the **Calculate number of documents to be processed** option to **No**.
4. Click **OK**.

Note

In some cases (such as external joins), the result that is returned may be corrupt.

Using the connector cache option

For Database-type connectors, or for those whose data format is stable, we recommend that you use a cache for the structure of the published document types (metadata).

The cache allows Database-type connectors to open more quickly since the published document-type description (metadata) is loaded locally. This is very noticeable when there are many tables or objects.

Note

In testing or development phases, the format of the data that is exchanged may change. In this case, we recommend that you deactivate the cache option.

If a connector does not use a cache, it has to obtain a data set corresponding to the source database description every time it opens.

For Database-type connectors, this description data includes the:

- List of tables.
- List of fields and their types.
- List of forms.
- List of indexes.
- List of links.
- List of joins.
- Etc.

The time needed by the connector to open increases with the amount of data.

Note

The time a connector takes to open is also slowed if available bandwidth is limited. This is linked to the performance of the network connection (WAN, LAN).

Warning

We recommend using a cache when you put a scenario into production.

However, if the description data of the source database changes, you must synchronize your connector's change.

To synchronize a connector's cache:

1. Open the Scenario builder.
2. Open your scenario.

3. Select the connector whose cache you want to synchronize.
4. Open the connector (F4).
5. Select the **Tools/Cache/Synchronize the cache** menu.

After synchronizing the cache, make sure that your mappings do not contain items that no longer exist in the new source database description.

Automatic reconnection

For database type connectors (ServiceCenter, Asset Manager, database, etc.), Connect-It's internal policy is to keep the connection with the server open in order to improve processing performance. These connectors can be reconfigured in order to reconnect to the server automatically if the connection is lost and after a predetermined delay.

If network stability is not maintained, reconnection may take a significant amount of time. Check the policy in place for your network regarding your network's firewall and planned or unplanned outages.

If you plan on using the connectors whose data source is a remote sever, you must select the automatic server connection option.

Automatic reconnection options have an impact on the network's latency as it multiplies the transmission of data-packets.

The reconnection information is displayed in the Connect-It log.

Example of the Asset Manager connector's attempt to reconnect:

```
Opening session for connector 'Asset Manager (TPRIM)'
Connecting to the server...
  'Asset Management' API error: 'Not connected to the Oracle server
  Unable to connect to database 'STRONTIUM' (UserId=TPRIM) ; Oracle code=12154
  Unable to connect to this database engine.'
  The next attempt to connect to the server will be performed in 1 s
Attempting to reconnect ...
  'Asset Management' API error: 'Not connected to the Oracle server
  Unable to connect to database 'STRONTIUM' (UserId=TPRIM) ; Oracle code=12154
  Unable to connect to this database engine.'
  The next attempt to connect to the server will be performed in 2 s
[...]
```

Close the connection to the server when the Connect-It session is closed

This configuration option enables you to close the connection at the end of each scenario session and to reconnect only when you open a new session.

This enables Connect-It server to use fewer resources and reconnect to the server more rapidly.

Warning

This option must not be selected if the scenario is executed frequently (for example, every five minutes).

Chapter 3

Optimization - Destination Connector

This chapter presents the different Connect-It parameters that impact the performance of a scenario.

Customizing the Target System

In addition to optimizing how Connect-It operates, you can also customize the target system. Remember to:

- Create indexes for the database.
- Avoid transferring files from one disk to another (use data on the same disk), especially in cases where the disks are accessed remotely.

Initial Load Options

This option is used for database-type connectors.

Before using this option, make sure the following prerequisites have been met:

- Start with an empty database
- The documents to import are unique

Selecting the initial load option lets you reduce the number of queries by limiting them to inserts only (INSERT). Reconciliation selection queries (preliminary checks to see if the documents exist) are not performed.

Using the Connector Cache Option

The behavior of this option is described in chapter Optimization - Source connector.

Automatic Reconnection

The behavior of this option is described in chapter Optimization - Source connector.

Reconciliation Keys

Reconciliation consists of defining the fields that enable the unique identification of the records in the tables that a connector creates or updates.

Fields used for data reconciliation are identified by marking them with a key, called the reconciliation key. Complex mapping elements may have multiple reconciliation keys. Each key belongs to a key set.

The field you use as a reconciliation key will greatly influence the consumption of documents by a destination connector.

When a scenario is executed, the following operations are performed:

- Creating the document type produced by the source connector
- The document type from the mapping is created
- Reconciliation is performed for the destination connector: The document's elements from the mapping are compared with the fields present for a given record in the database
- Reconciliation scripts are applied for the destination connector

How reconciliation keys work

Reconciliation consists of defining the fields that enable the unique identification of the records in the tables that a connector creates or updates.

Consuming documents involves the following two actions:

1. Sending a query that uses the fields selected as reconciliation keys to learn if the record exists in the destination application.
2. Inserting or updating the record. If the fields selected as reconciliation keys are not indexed, it will take longer to search through the database and consume documents.

Processing performance is increased when:

- Indexed queries are used to retrieve a record
  Unique constraints are used to optimize key identification (key set weight)
- For example, to retrieve records from the departments and employees table, you can use an employee's ID instead of the employee's last name, first name and telephone number.

Improving the production of documents

Improving a connector's document consumption involves processing, as rapidly as possible, data to be sent to a destination application.

This section explains the different ways you can reduce the time it takes to consume documents.

Transaction management

By default, the database-type connectors perform a 'commit' each time they consume a document.

You can perform the 'commit' by group of documents.

To set the number of documents to consume before committing them:

1. Open the Scenario builder.
2. Open your scenario.
3. Select the connector for which you want to modify the transaction parameters.
4. Launch the connector configuration wizard (F2).
   Warning: Verify that you are in advanced configuration mode (the icon is selected in the toolbar).
5. Click **Next** several times until you come to the **Manage transactions** page.
6. Select the **Commit by group of documents** option.
7. Indicate how many documents should be consumed before a 'commit' is triggered.
8. When numerous documents are consumed, we recommend that you launch the 'commit by group' function for 200 documents or more.

For example, for a given scenario, processing performance is as follows:

- Commit after each document (default option)
  Performance: Four minutes for 6,000 documents, or 25 documents per second.
- Commit by group of 500 documents
  Performance: Three minutes for 6,000 documents, or 33 documents per second.

Commit

Based on our research and depending on each scenario, we found that a 30% improvement was achieved when configuring the commit by batch (between 20 and 100 records) as compared to an immediate commit. An improvement in performance is directly related to the object of the transaction and the transaction locks.

This explanation is only for unique injectors. When you use multiple injectors that access the same data, the number of server locks will increase and may create longer commit delays. This will have a negative impact on processing performance because the other injectors will need to wait until the locks are released before accessing the data.

Also, this may result in a situation where a perpetual deadlock persists. In this case, one of the applications is selected and cancelled by the DBMS.

Batch processing

If batch processing is selected, when a document generates an error, the entire set of documents is retried one by one. In this instance, batch processing can be more costly than document by document processing if the batches contain a large number of documents.

We recommend that you make batches of 10 or 20 documents in order to prevent this behavior (possible errors).

Parallelization

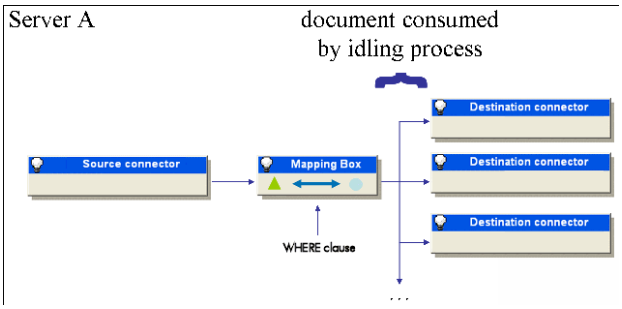
Implementing parallelization in a scenario is done to bring answers to the following question:

How can I maximize the use of multiple processors?
Objective: Execute multiple instances of a connector in order to increase processing times.

Using parallelization for a scenario means:

Assign a number of processors to execute à scenario.
Testing can be performed to ensure that the different processes can run at the same time without incident.

Parallelization - how it works



When parallelization is used, several processes run at the same time. A parent process controls its child processes and triggers them when possible.

Example: Three processes are executing in parallel. The processes are composed of one parent process and two child processes. The parent process uses 50% of the processor. The parent process can trigger one of its child processes which will run using the remaining processor power, 50% in this case. Consequently, when the third process that is waiting is launched, available processor time is reallocated and the performance of each connector is reduced by one-third. In this case, there is no need to have a third process which, because of its behavior, would block processing gains.

Note

Processing performance during parallelization is directly linked to the way in which documents are consumed and produced.

For a dedicated machine, the ideal processor power consumption optimization is around 90 to 95%.

Also, you should avoid, as much as possible, the allocation of virtual memory as it is source of increased disk access times.

Launching parallel scenarios

When one scenario must migrate large amounts of data, you sometimes need to divide it into several scenarios and launch each one on a different service (Windows) or daemon (Unix). Also, it may be useful to launch the services on different servers.

Example

You need a scenario to import the list of employees recorded in one database to another database.

To improve the time it takes to migrate this data, you can create two scenarios:

- The first scenario migrates the list of employees whose name starts with a letter between A and J.
- The second scenario migrates the list of employees whose name starts with a letter between K and Z.

Employees are selected when you enter production directives (WHERE clause) for your two scenarios' source connectors.

Locks

This section involves database type connectors.

The goal is to find the right balance between server locks and the time required for processing.

The main disadvantage of processing data in parallel occurs when records are locked by the server while they are being processed. Locking leads to a decrease in processing performance.

As the role of each Connect-It process is to consume a document type, in some circumstances, there may be several concurrent accesses to the same table for a given scenario.

Records are locked by the server while it is being created or updated. If another process tries to access the record, the process is notified that a lock exists on that record. This lowers performance as the process cycles through a series of attempts to obtain access to the record.

This slows scenario execution and lowers consumption.

Waiting for lock removal

When a record is locked by the server, the process trying to access that record is instructed to wait. The process then enters a state where it waits and then tries to gain access to the record at regular intervals.

This mode of operation has a drawback: the process may still be in wait mode even after the record has been unlocked. In this instance, the parent process (or another process) may access the same record and lock it again. The other process then tries to access the record and is again instructed to wait. If the process has to wait too long, it returns an error.

Avoiding deadlocks

Several approaches are possible to avoid record locking:

- Records can be locked only when the process is in write mode. For a given scenario, you can differentiate between processes that insert and update data, and processes that need only to read data.

- Structure the scenario to have processes query records in read mode and not in write mode. For example, create the relevant records before they are queried by the processes. For an HP Asset Manager application, the models table is inserted before the data set that references it.

Chapter 4

Optimization - Engine

This chapter presents the different Connect-It parameters that impact the performance of a scenario.

Using Connect-It

Improving document processing times of the Connect-It engine

This section explains what you can do to reduce the time it takes for the Connect-It engine to process documents.

Using the non-graphical interface

If the Scenario builder is displayed while the scenario is running, the Connect-It engine will react more slowly each time you choose a command: refresh, consult the document log, enter a new option, etc.

Setting the scheduling

Scheduling in Connect-It is used to associate one or more schedulers to document types produced by a scenario.

Example: A scheduler wakes up the Asset Manager connector every hour. Each time it wakes up, the connector produces documents corresponding to records in the Asset Manager application it consults.

Each time the connector wakes up, Connect-It calculates how much data needs to be processed and does not stop until all the data has been processed. You must set your schedulers depending on how much data there is to process. When migrating large databases, we recommend that you program your scheduler to wake up the connector during periods of little or no activity (at night, for example). For integration scenarios that will process less data, we recommend using the predefined **Synchronous** scheduler (which wakes up the connector every second).

Start a Scenario

Customizing the .cfg file

The .cfg file can be customized using the configuration file editor.

You can customize the .cfg file to create a view above the flat view of the produced document type (table and fields).

You can also view information related to several tables in a single document and create new joins or sub-queries. The drawback of this approach is that creating a unique, comprehensive document requires the creation of numerous queries and other costly database operations that can negatively impact performance. For example, the database connector has a read capacity of 2,000 documents per second, whereas the IntelLanDesc connector (which was developed using the database connector to which a metadata description was added) has a read capacity of approximately 7 documents per second.

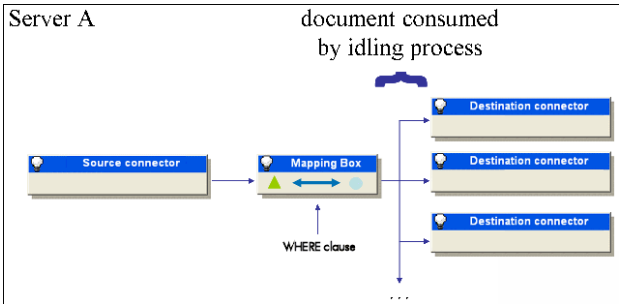
Injectors

An injector is any object (scenario, connector) that injects data.

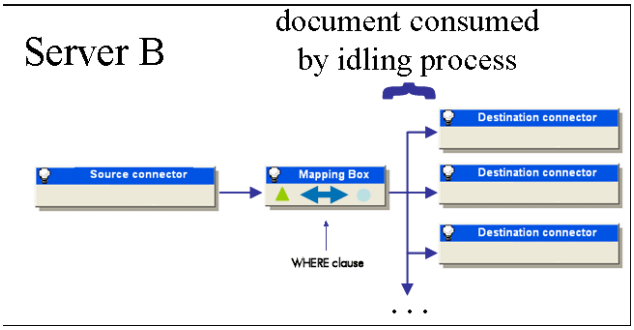
In order to increase the processing performance of a scenario, the following options can be used:

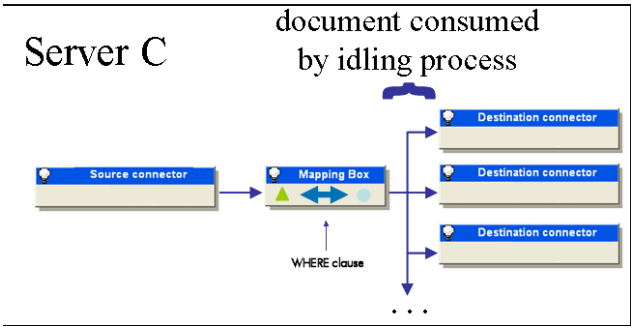
- Launch multiple instances of the same scenario
- Launch multiple instances of the same scenario

Example: 3 injectors run 3 processes in parallel, which results in injecting 9 data sources simultaneously.



The scenarios running the 3 injectors must not process data at the same time. A filter based on a WHERE clause must be used for each injector to ensure that data is loaded and processed once only.





Records Cache

Caching lets you reuse information retrieved by previous queries. If a query has already been sent requesting a record ID, a subsequent query requesting the same information will not be sent to the data source. The record's ID is retrieved from the Connect-It cache database. This behavior can be set via the Options/ Component/ Maximum number of documents in the reconciliation cache menu.

The next time the scenario is used, Connect-It will substitute its identifier in the cache rather than checking for the object in the database. This saves time as the query does not have to be sent to the database. This option is useful for common cases where frequently used tables are analyzed, such as the departments and employees, products, catalog references, and other tables.

The default size of the cache is 1,000 records for a table associated with a node in a document type.

Caching provides a 30% gain for SQL Select type queries for a connector during basic testing. This means a 30% savings in transactions with the database server.

Configuring the option for the maximum number of documents saved in the reconciliation cache has an impact on processing performance.

The configuration of the cache provides advantages, but can produce some drawbacks under certain circumstances.

- Advantage: Limits the number of roundtrips between the connector and the destination database.
- Drawback: If the cache is too large, resources available on the server will be limited.

Note

Do not confuse the records cache and the connector cache (defined when the connector is configured via the configuration wizard), which corresponds to a snapshot of the database structure queried by the connector at a given time.

Connect-It User's Guide - Using cache files

Connect-It Connectors Guide - Configure the cache (advanced mode).

Scripts

- Limit the number of variable declarations in scripts
  For example, the following script:

  ```
  dim test1,test2,test3 as string
  test1="abcd"
  test2="efg"
  test3=test1+test2
  ```

  has a greater impact than this script:

  ```
  dim test as string
  test="abcdefg"
  ```

  Also, it is better to declare variables in specific files containing other global variables.

- Remove data from programming loops where possible.

Log

Using the Document log enables you to see if any problems occurred while processing the documents produced by a source connector.

The following actions will increase processing performance:

- It is recommended to configure the document log to only log documents that create processing problems.
- You will need to define the number of lines to load into memory depending on the resources available on the server that is managing the scenarios.
  If a large number of documents are saved to memory, all available physical memory may become saturated. Consequently, the server will need to create virtual memory which will increase processing times.

Configuration

Configuring the document log

Configuring Connect-It's Document log enables you to indicate:

- The error type that you want to obtain in the Document log (**Filter** field).
- The maximum number of tracking lines that Connect-It saves in memory.
- A text file in which the messages from the Document log will be saved.

To improve the time it takes to process documents:

1. Launch the Scenario builder.
2. Select the **Monitors/ Configure monitors...** menu, then click the Session backups tab and disable:
   - Database
   - Files
3. Limit the error type that you want to save in the Document log by selecting a sub-option for the **Save documents** option.
   Example: Reject.
4. Limit the number of tracking lines to be saved (**Display sessions** tab)

Configuring the monitors

Monitors let you define monitoring functions associated with a scenario.

Performance may vary depending on the filters that are applied. Special attention must be given to:

- The maximum amount of allocated memory.
- The number of tracking lines saved in the document log.
- The filter that is used on the document (reject, field reject, etc.).

Statistics

Evaluating document-processing times using statistics

Connect-It manages statistics relative to different elements that make up a scenario.

These statistics are used to view the time required to process data and to locate any bottlenecks, especially those which may occur when producing or consuming irrelevant documents.

The Connect-It logs, which you can consult in the graphical interface of the Scenario builder or in the log files, provide statistics about:

- The time it takes the source connectors to produce documents.
- The time it takes the mapping boxes to transform documents.
- The time it takes the destination connectors to consume documents.
- The number of rejected documents.
  Example: The statistics for the Asset Manager connector inform you that the documents corresponding to the records in the Products table were not processed.

to obtain statistics on how a tool processes data

1. Open a scenario
2. Configure the monitors (**Monitors/ Configure monitors** menu)
3. Open the scenario connectors (**Ctrl + F4**).
4. Select the scenario's source connector.
5. Click ▸ (**F5**).
6. Position your pointer on a source connector, a destination connector or a mapping box.
   A contextual window appears containing a **Statistics** section.

   Note:
   This data is also available in the Connect-It log tab, which is located under the scenario diagram.

Analyzing the statistics

The following data - displayed in the Connect-It log - represents the statistics of a scenario using an Action Request System connector as its source, an LDAP connector as its destination, and a mapping box:

```
Statistics for the 'Action Request System (fdcitsrv01)' connector  (session: 26m
in  57.310s / API: 26min  04.550s)
       Document(s) consumed: 7143
       Document(s) rejected: 1
       Records(s) inserted: 1199
       Records(s) updated: 5943
 Statistics for the 'LDAP (mail-sd.Hewlett-Packard.com)' connector (session: 47.
628s / API: 35.765s)
       Document(s) produced: 7143
 Statistics for the 'Mapping (Basic engine)' connector (Session: 01.404s)
 Script(s) analyzed: 14286
       Document(s) consumed: 7143
       Document(s) produced: 7143
```

The following is an analysis of the produced statistics:

- Document-processing time of the Action Request System connector = 27 minutes, or **4.4** documents per second.
  The processing time breaks down into 26 minutes for the APIs concerned by the Action Request System connector and 1 minute for processing in Connect-It.
  APIs include the network response time, the execution of 'commits', respecting the database integrity rules, etc.

- Document-processing time of the LDAP connector = 48 seconds, or **150** documents per second. The processing time breaks down into 36 seconds for the LDAP API and 12 seconds for processing in Connect-It.
- Document-processing time of the mapping box = 2 seconds, or **5100** documents per second. This duration corresponds to the document-processing time in Connect-It.

Advanced statistics

- Advanced statistics are available for database type connectors. They enable you to:
  - View the time used for Select, Insert, Update and Delete type queries.
  - View minimum, average and maximum times, as well as the total number of Select, Insert, Update and Delete queries that were issued.

Examples of document-processing speeds

Here are a few examples of different document-processing speeds in production:

- Database connector
  1500 to 2000 documents/second.
- ServiceCenter connector
  450 documents/second
- Asset Manager connector
  400 documents/second

Database Connector

Improving Performance of a Scenario Using a Database Type Connector with Sybase

For the Database type connectors that use (if needed) a native Sybase connection, you can improve the execution of SQL queries by entering the `PostConnectSql=set forceplan on` option in the advanced options.

Under Sybase 12 and 12.5, unique indexes are taken into account and non-unique indexes are ignored.

Customizing the Database

Several approaches can be adopted to improve processing speed:

- Indexes
- Statistics
- SQL query structures

Indexes

It may be beneficial for the index to cover multiple fields that will be queried rather than have the index cover each field individually. In this case, the index covers a column instead of a row.

For example, for Oracle, a Select type query on several columns has the following structure:

```
SELECT W1.lWorkItemId, W2.lWfInstanceId FROM amWfWorkItem W1, amWfInstance W2
WHERE W1.lDocRecordId = 28836763 AND W1.lActivId = 1417 AND W1.seStatus = 0 AND
W2.seStatus = 0 AND W1.lWfInstanceId=W2.lWfInstanceId;
```

Processing time: 00:00:02.02

If this query is executed 10,000 times to process all documents, total processing time will be 20,200 seconds, or 5.6 hours.

```
Statistics
----------------------------------------------------------
    12400  consistent gets
    12389  physical reads
      185  bytes sent via SQL*Net to client
      234  bytes received via SQL*Net from client
```

By adding the following index:

```
create unique index workwfitem_20051124 on
amwfworkitem(ldocrecordid, lactivid, sestatus, lwfinstanceid, lworkitemid);
```

Processing time: 00:00:00.07

```
Statistics
----------------------------------------------------------
      3  consistent gets
      2  physical reads
    185  bytes sent via SQL*Net to client
    234  bytes received via SQL*Net from client
```

The savings for the DBMS equals 3,000.

In total, creating the multi-column index provided a savings of 19,300 seconds, or 5.3 hours.

Statistics

In most cases, you will need to update the table statistics via your DBMS.

For example, using Oracle, you must update the statistics if the SQL query optimizer mode (optimizer_mode) is ALL_ROWS (or FIRST_ROWS), for each table.

Statistics are not required if the optimizer mode is set to RULE.

You can do the following using SQL PLUS on Oracle to calculate statistics:

```
ANALYZE TABLE xxx COMPUTE STATISTICS;
```

SQL query structures

In general for Oracle, if sub-queries are used, make sure you are using the EXISTS or NOT EXIST instructions instead of IN.

The EXISTS instruction has the following behavior: If the value returned by the query is TRUE, then the value TRUE is set for the entire operation. This prevents you from having to perform other time-consuming queries.

Note

This behavior is not relevant on DB2 and MSSQL Server which can benefit from other optimization measures.

Asset Manager Connector

Improving performance of a scenario using the Asset Manager connector

This section explains the different ways you can reduce the time it takes to process documents in a scenario using the Asset Manager connector.

Indexing the dtLastModif field

In a scenario using the Asset Manager connector in scheduled mode, you must add indexes to the **dtLastModif** field in all Asset Manager tables covered by the scenario, if this field is not already present. This field is also used systematically by Connect-It to verify the records created or modified since the last session.

To create an index on the Assets table (amAsset), Products table (amProduct) or Employees and Departments table (amEmplDept), execute the following command:

```
CREATE INDEX Ast_dtLastModif ON amAsset (dtLastModif)
GO
CREATE INDEX Prod_dtLastModif ON amProduct (dtLastModif)
GO
CREATE INDEX EmplDept_dtLastModif ON amEmplDept (dtLastModif)
GO
```

To verify that the **dtLastModif** fields were all indexed, you can use the adblog executable. This executable enables you to verify if the SQL queries that have a filter on the **dtLastModif** field in the WHERE clause, were executed.

Settings for an Asset Manager database - Sybase ASE engine

If the LOG files indicate that certain queries are taking time and that they have numerous tables in their FROM part, we recommend that you add the following line in order to reduce processing time.

```
PostConnectSql= set forceplan on
```

Modifying the amdb.ini file

To reduce the time it takes to process SQL queries, you must also modify the **amdb.ini** file in your Asset Manager application by adding the following line:

```
PostConnectSql= set forceplan on
```

The following lines show the configuration of the **amdb.ini** file for a database called **DB ASE COPPER**:

```
[DB ASE COPPER]
PostConnectSql=set forceplan on
stmtcache=500
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC6
41ED76CDAA17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

These parameters must be configured in the **amdb.ini** file on the Asset Management database server.

If the client part of your Asset Manager application is installed on your Connect-It server, you can establish two different connections linked to the same Sybase database:

- The first connection uses the `PostConnectSql=set forceplan on` and `stmtcache=500` options.

  ```
  [DB ASE ConnectIt]
  PostConnectSql=set forceplan on
  stmtcache=500
  LongDesc=
  ```

```
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC
641ED76CDAA17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

- The second connection does not use either of these parameters.

```
[DB ASE COPPER ACGUI]
LongDesc=
Engine=Sybase
Location=COPPER
EngineLogin=itam
EnginePassword=78C6143D43925F46F924205FBB42F0FED21594428DDCAC641
ED76CDAA17050EA1A124254200200
ReadOnly=0
CacheDir=
CacheSize=5120000
Base=EDS
Owner=
TableSpace=
TableSpaceIndex=
AmApiDll=aamapi35.dll
UseNTSecurity=0
```

For the classic Asset Manager clients (other than the Sybase ASE clients), the `stmtcache=500` option must not be used.

If you encounter performance problems on the classic Asset Manager clients, you can try one of the following options:

- PostConnectSql=set forceplan on
- PostConnectSql=set table count 3
- PostConnectSql=set table count 2

Chapter 5

How to Improve the Performance of a Scenario - Example

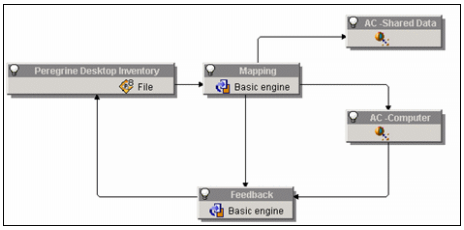
This chapter explains how to modify a scenario in order to increase processing performance.

Example

A Connect-It scenario populates an Asset Manager database with inventory data.

The scenario manages:

- Models (hardware and software) in one mapping. A single injector is used.
- Hardware and software installations in a second mapping. Multiple injectors are used to link the installations to models in order to reduce server locking.

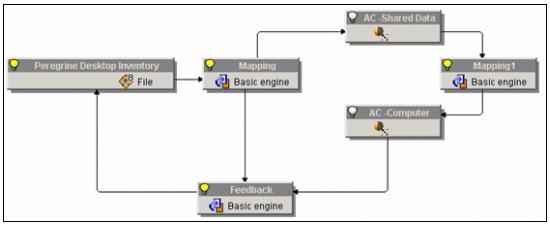


Test results: Imported 10,000 hardware records and 23,783 total records in 54 minutes

Test results (second test): Imported 10,000 hardware records and 807,378 total records in 8 hours and 40 minutes

The following changes can be made to increase performance:

- Documents are only produced once by the source connector instead of twice as initially planned.
- Use a new behavior that creates the models first, then inserts data after they are created.



Test results: Imported 10,000 hardware records and 23,783 total records in 47 minutes

Gain: 13%.

Test results (second test): Imported 10,000 hardware records and 807,378 total records in 6 hours.

31% gain.

The following conclusion can be made: If data must be processed several times in different mappings, it is preferable to produce the data only once and then to refine it rather than producing the data several times.

Index