

Technical Note

Managing MXCS and the Module File Cache with SQLXPress

Document Version 1.00

Copyright ©2013 Merlon Software Corporation



.....	1
Technical Note.....	1
Introduction.....	4
Background.....	4
MXCS Management and Configuration.....	5
Manage MXCS Services.....	5
To Start a Service.....	6
To Stop a Service Abruptly.....	6
To Stop a Service Gracefully.....	6
To Delete a Service.....	7
To Create a New Service.....	7
Manage MXCS Datasources	7
To View The Detailed Properties of a Datasource.....	8
To Add a Datasource.....	8
To Alter a Datasource.....	8
To Enable or Disable Datasource Statistics Collection.....	8
To Start a Datasource on All Services.....	8
To Stop a Datasource on All Services Abruptly.....	8
To Stop a Datasource on All Services Gracefully.....	8
To Start a Datasource Trace on All Services.....	9
To Stop a Datasource Trace on All Services.....	9
To Delete a Datasource From All Services.....	9
Manage MXCS Users.....	9
To Add a User.....	9
To Delete a User.....	10
To Alter a User.....	10
Extract MXCS Statistics from EMS.....	10
Display MXCS Statistics.....	11
Module File Cache Management.....	15
Extracting MFC Execution Plans	15
Browsing MFC Execution Plans.....	16
Finding MFC Execution Plans with specific Criteria	17
Locating Inefficient Plans or Ineffective Plans.....	17
Determining the Effect of Changes to the Database or to Control-Query-Defaults.....	19

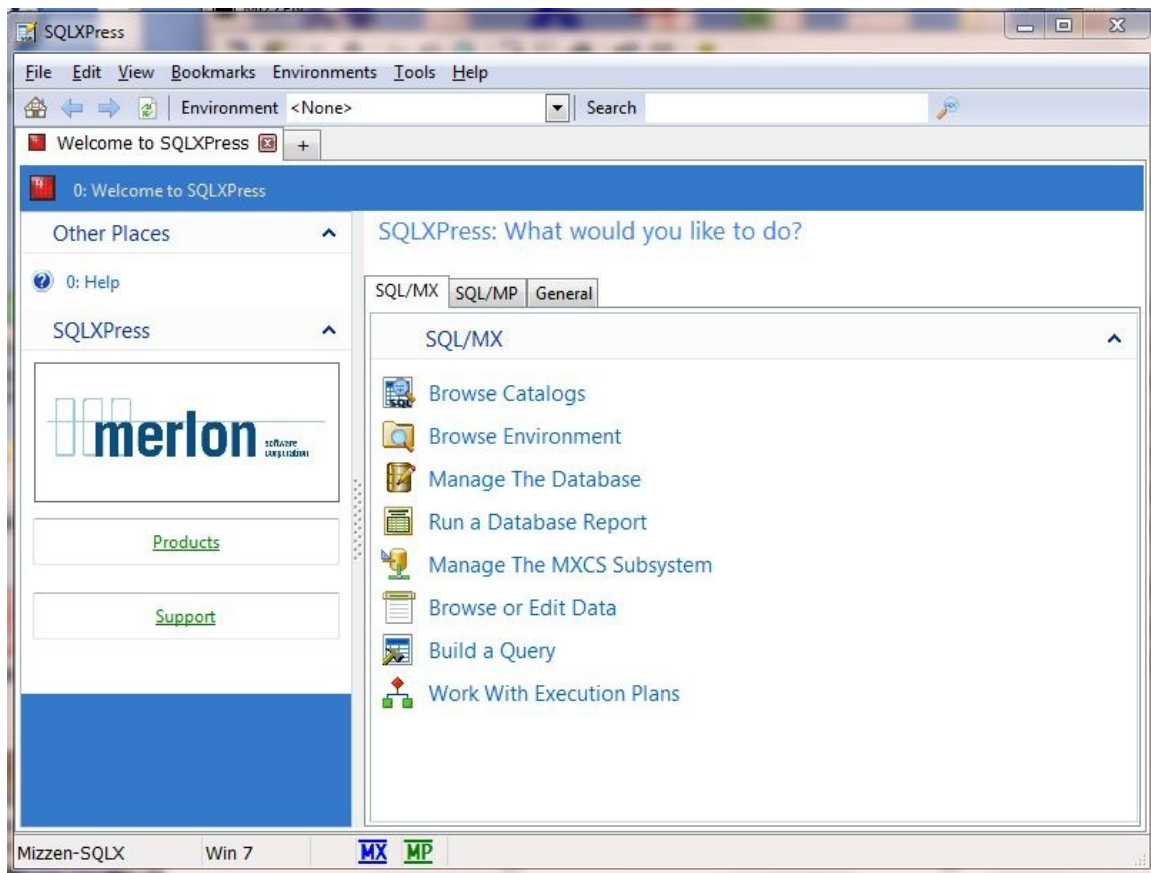
Introduction

This document explains the MXCS (SQL/MX Connectivity Services) and Module File Cache management facilities in Merlon's SQLXPress. Most of these features are available in release SQLXPress 3.21, although some Module File Cache report tools were introduced in the 3.30 release of SQLXPress.

Background

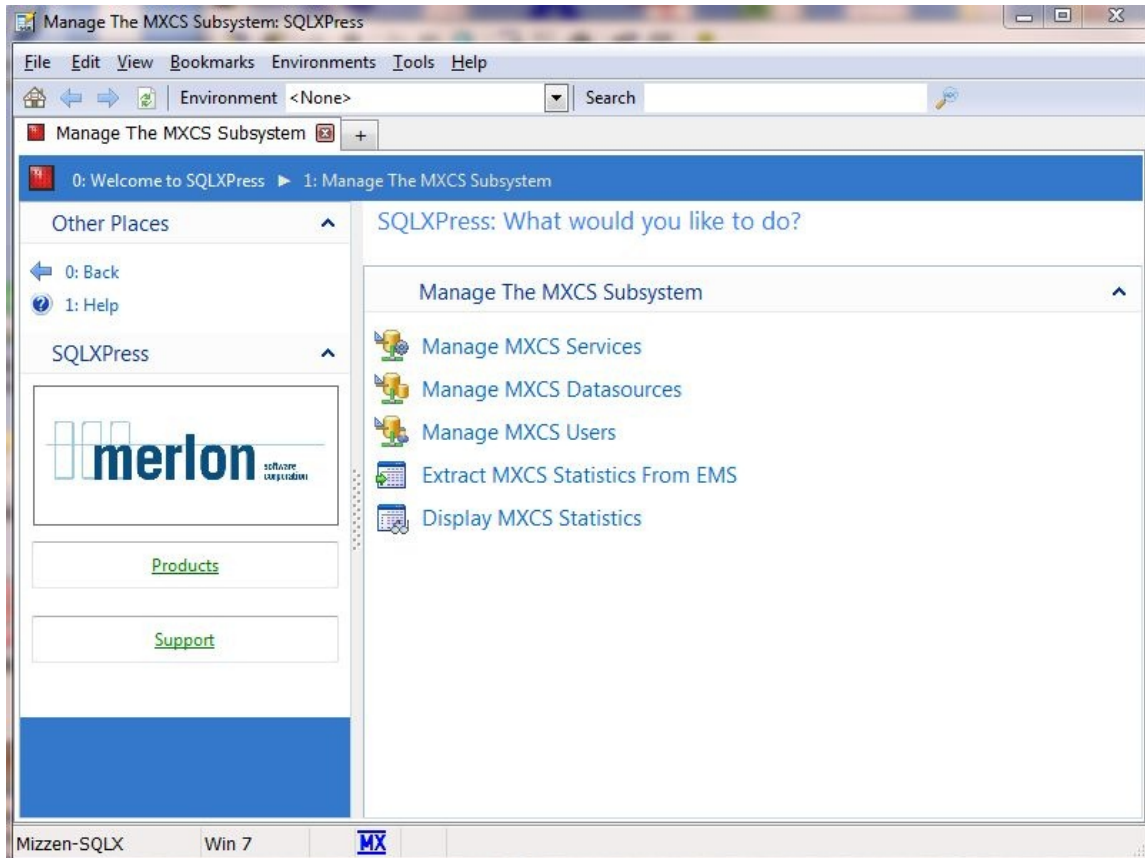
MXCS consists of a set of services and servers, as well as some other configuration information, which allow remote computers to connect to the SQL/MX database using ODBC or JDBC client programs. A Module File Cache is an area on the NSK system where plans for SQL queries are "cached", that is, saved so that if a subsequent query that is identical to one that has already been compiled and executed occurs, it can be executed from the cached plan without requiring a potentially expensive compilation. The Module File Cache (MFC) can significantly improve the performance of the NSK database, but should be carefully monitored and maintained: out-of-date MFC entries can cause performance problems if the plan used is out of sync with the current state of the database.

SQLXPress is a client-server application. The client runs on a PC workstation and sends commands to a server program that runs on the NSK system. The SQLXPress client provides services that allow a Database Administrator (DBA) to configure and control the MXCS and MFC components of SQL/MX. Both features are accessed from SQLXPress home window, shown below:



MXCS Management and Configuration

Choose the option “Manage the MXCS Subsystem on the home window. The SQLXPress Manage MXCS Subsystem window will appear:



Manage MXCS Services

This page displays a list of the Association and Configuration servers for the MXCS services on the NonStop server.

Each service instance has one Association server and one Configuration server.

From here you can start and stop a service.

To create a new service, click on the Create MXCS Service link in Related Tasks.

You can also delete a service which will abruptly terminate the association and configuration servers as well as any datasource servers.

The grid lists details for each Association and Configuration server.

The following are displayed:

- **Process Name.** The process name of the server.

- **State.** The state of the service.
 - **Available:** The service is running and is available for use. ODBC and JDBC clients may connect to datasources in this service.
 - **Starting:** The service is initializing.
 - **Stopping:** The service is shutting down. Some datasource servers are still running.
 - **Stopped:** The service is running, but is not accepting any new connection requests.
- **Server Type.** The type of server, this is either Association or Configuration.
- **CPU.** The number of the CPU where the process is running.
- **PIN.** The Process Identification Number of the process.
- **TCP/IP Process Name.** The name of the TCP/IP process that the service process is using for network communications. The IP address of the TCP/IP process must be used by clients wishing to connect to the service.
- **TCP/IP Port Number.** The port number used by the process. To connect to a service, a client must be configured to use the port number of the Association server.

To Start a Service

- Select an Association server in the grid, and click Start.
- In order to use this command the service should be in the Stopped state.
- The service must have been previously created.

To Stop a Service Abruptly

- Select an Association server in the grid, and click Stop Now.
- The service will stop immediately.
- Active client sessions will be abruptly disconnected.
- In order to use this command the service should be in the Available state.

To Stop a Service Gracefully

- Select an Association server in the grid, and click Stop After Disconnect.

- The service will stop accepting new connection requests.
- Active sessions will continue until they are disconnected by the client.
- The service will be placed into the Stopping state until all clients have disconnected, at which point the service will enter the Stopped state.
- In order to use this command the service should be in the Available state.

To Delete a Service

- Select an Association server in the grid, and click Delete.
- This will abruptly stop the Association server, the Configuration server, and all datasource servers for the service.
- It is recommended that you stop the service before you delete it.

To Create a New Service

- Click on the Create MXCS Service link in Related Tasks.
- You will be asked to specify the properties of the new service. A new Association server will be started which will complete the creation of the new service.

Manage MXCS Datasources

This page displays a list of MXCS datasources.

From here you can start a datasource, stop a datasource, start a trace, stop a trace, and delete a datasource.

You can also add or alter a datasource.

There are two aspects to an MXCS datasource - configuration, and status.

An MXCS datasource has the same configuration in all MXCS service instances. So for example, you can't set its MaxServers property to 100 in one service, and 200 in another service. Datasource configuration values are global across services.

On the other hand, the status of a datasource may be different in each MXCS service instance. For example it may be stopped in one service, and started in another. Datasource status is local to each service.

Commands you issue from this page apply to the datasource in all services. If you want to manage a datasource for a particular service, use the MXCS Datasource Status page instead

The grid contains a row for each datasource. It shows a summary of the properties of each datasource.

To View The Detailed Properties of a Datasource

- Select a datasource in the grid.
- Click the Datasource Properties link in Other Places.

To Add a Datasource

- Click the Add Datasource link in Related Tasks. This wizard will step you through the process of adding a datasource - it will be added to all services.

To Alter a Datasource

- Click the Alter Datasource link in Related Tasks. This wizard will step you through the process of altering a datasource - it will be altered in all services.

To Enable or Disable Datasource Statistics Collection

- Click the Alter Datasource Statistics Options link in Related Tasks. You can enable or disable statistic collection from this page.

To Start a Datasource on All Services

- Select a datasource in the grid, and click Start. The datasource will be started on all services.

To Stop a Datasource on All Services Abruptly

- Select a datasource in the grid, and click Stop Now.
- The datasource will stop immediately on all services.
- Active client sessions will be abruptly disconnected.

To Stop a Datasource on All Services Gracefully

- Select a datasource in the grid, and click Stop After Disconnect.
- All services will stop accepting new connection requests for this datasource.
- Active sessions will continue until they are disconnected by the client.
- The datasource will be placed into the Stopping state until all clients have disconnected, at which point the datasource will enter the Stopped state.
- The datasource may not enter the stopped state at the same time for each service. Use the MXCS Datasource Status page to check the datasource status within each service.

To Start a Datasource Trace on All Services

- Select a datasource in the grid, and click Start Trace.
- Tracing will be started for datasource servers on all services. Trace data is written to the EMS collector specified in the Association server's startup command via the -TMC parameter, or \$0 if the -TMC parameter was omitted. (Each service instance may log to a different EMS collector).

To Stop a Datasource Trace on All Services

- Select a datasource in the grid, and click Stop Trace.
- Tracing will be stopped for all datasource servers.

To Delete a Datasource From All Services

- Select a datasource in the grid, and click Delete.
- The datasource must be in the stopped state in all services, or an error is returned.

Manage MXCS Users

This page is used to configure user permissions for the MXCS subsystem

Users can be registered in the MXCS configuration database.

Each user is assigned a permissions value of either *User* or *Operator*.

A User is allowed to view the configuration and state of MXCS, but cannot make any changes.

An Operator can change as well as view the configuration and state of MXCS.

This page lists all of the users that have been defined to MXCS.

The special user called "Public" is defined as a User.

The super id SUPER.SUPER is always defined and an Operator.

Only the super id SUPER.SUPER can modify user permissions. This means that you must be logged on as SUPER.SUPER to apply changes using this page.

The grid displays the current set of user permissions.

To Add a User

- Click Add, a row is added to the grid.
- Enter a user name, or click the button to select a user from a list.
- Select a permissions value from the drop down list.
- Click Apply, the user will be added.

To Delete a User

- Click on the user name in the grid.
- Click Delete, the row will be removed from the grid.
- Click Apply, the user will be deleted.

Note that you cannot delete the Public or SUPER.SUPER users.

To Alter a User

- Edit the user name and / or permissions value in the grid.
- Click Apply.

If you alter the user name and click Apply, the existing user entry will be deleted from the configuration database, and the new user will be added.

You can make multiple changes before clicking Apply, the page will issue the appropriate set of add, alter, or delete commands to implement the changes you have made.

Extract MXCS Statistics from EMS

You can configure a datasource to write statistics to an EMS collector.

Use this page to extract MXCS statistics event messages from EMS, and write them to the SQLXPress database. After extractions has been completed, use the Display MXCS Statistics page to display the statistics.

When you start the EMS extraction process from this page, any existing statistics in the SQLXPress database will be purged before the new statistics are captured in the database.

You can save a copy of the existing statistics by going to the Display MXCS Statistics page and using the Export to PDF or Export to Excel commands before you run the extraction.

To extract statistics, proceed as follows:

- Select the EMS collector to which the MXCS service has written its statistics events.

Note:

The statistics EMS collector is specified for a service when the service is created by running the Association server.

The EMS collector name is then passed to each datasource server process in its startup message.

If you don't know which EMS collector to use, you can find out as follows:

- Find the process name of a datasource server process that is running in the service you are interested in. Any server will do, you can view a list of servers in the MXCS Server Status page.

- Logon to TACL and issue a status, detail command on the server process, like this:

status \$X43X,detail

- In the output from the status command, look for the OSS Arguments value. It contains a parameter called -RMC which is followed by the EMS collector name used to log statistics events. Select this process name from the EMS collector drop down list.
- If you are interested in the statistics for a particular datasource server process, enter its name in the MXCS server process field, otherwise leave it blank to get statistics for all servers.
- Select the time at which to start scanning the EMS log for MXCS statistics events. You can start scanning at any time within the last two weeks. The earlier the start time, the longer the extraction is likely to take.
- Specify when the extraction process should stop. there are three choices:
 - At the end of the EMS log.
 - At a specified time.
 - When you click the Stop button. If you choose this option, extraction will continue even after reaching the end of the EMS log. The extraction will continue to monitor the EMS collector for new events.
- Click Start, the extraction process will be started.

When the extraction has been completed, the Display MXCS Statistics page will be opened so that you can view the statistics.

Display MXCS Statistics

Use this page to display MXCS statistics that were previously extracted from an EMS collector using the Extract MXCS Statistics from EMS page.

There are two tabs on this page.

The Query Statistics tab displays details statistics for SQL queries executed on behalf of ODBC/JDBC clients.

The Session Statistics tab displays summary statistics for client sessions.

Query Statistics

These statistics are enabled by setting the SQLxxx stats flags for a datasource.

The amount of data collected for a query depends on which of the SQLxxx flags have been turned on. To get the most information, we suggest that all of the flags be turned on.

Some queries may have incomplete information. This can happen if the extraction process started or stopped midway through the execution of a query for example.

Executing a single SQL query can generate many EMS events, each of which contains various pieces of information. On this page, these events are consolidated into a single record for each query.

The grid at the top of the tab contains a row for each query for which there are statistics.

The grid contains this information for each query:

- **Server.** The name of the datasource server process that executed the query.
- **Time.** The time at which the query was executed.
- **Client.** The name of the remote system that submitted the query.
- **Datasource.** The name of the datasource to which the server belongs.
- **Operation.** The operation type. this is one of SQLPrepare, SQLExecute, SQLExecuteDirect, or Fetch.
- **Statement ID.** The statement name (sometimes referred to as the cursor name).
- **Statement Type.** The type of SQL statement, for example insert, update, delete, or select.
- **Elapsed Time.** The time that elapsed between the start and the end of the query.
- **Compile Time.** The time taken to perform the SQL compilation of the query.
- **Execution Time.** The CPU time taken to execute the query.

The grid is initially sorted by server name and time.

You can change the sort order by clicking on a column heading.

For example, click the Execution Time column to sort in ascending execution time order. Click it again to sort in descending order.

In this way you can easily identify the queries that took the longest time to execute.

When you select a row in the top grid, the panel bellow the grid is updated with details about the query in the selected row.

The contents of the Details grid will vary depending on the type of query, and which stats flags were enabled.

The Query Text tab shows the SQL statement text (the SQL Statement stats flag must be enabled to get the complete statement text).

Session Statistics

These statistics are enabled by setting the connection and session summary flags for a datasource.

Some sessions may have incomplete information. This can happen if the extraction process started or stopped midway through a session for example.

The grid contains the following information for each session:

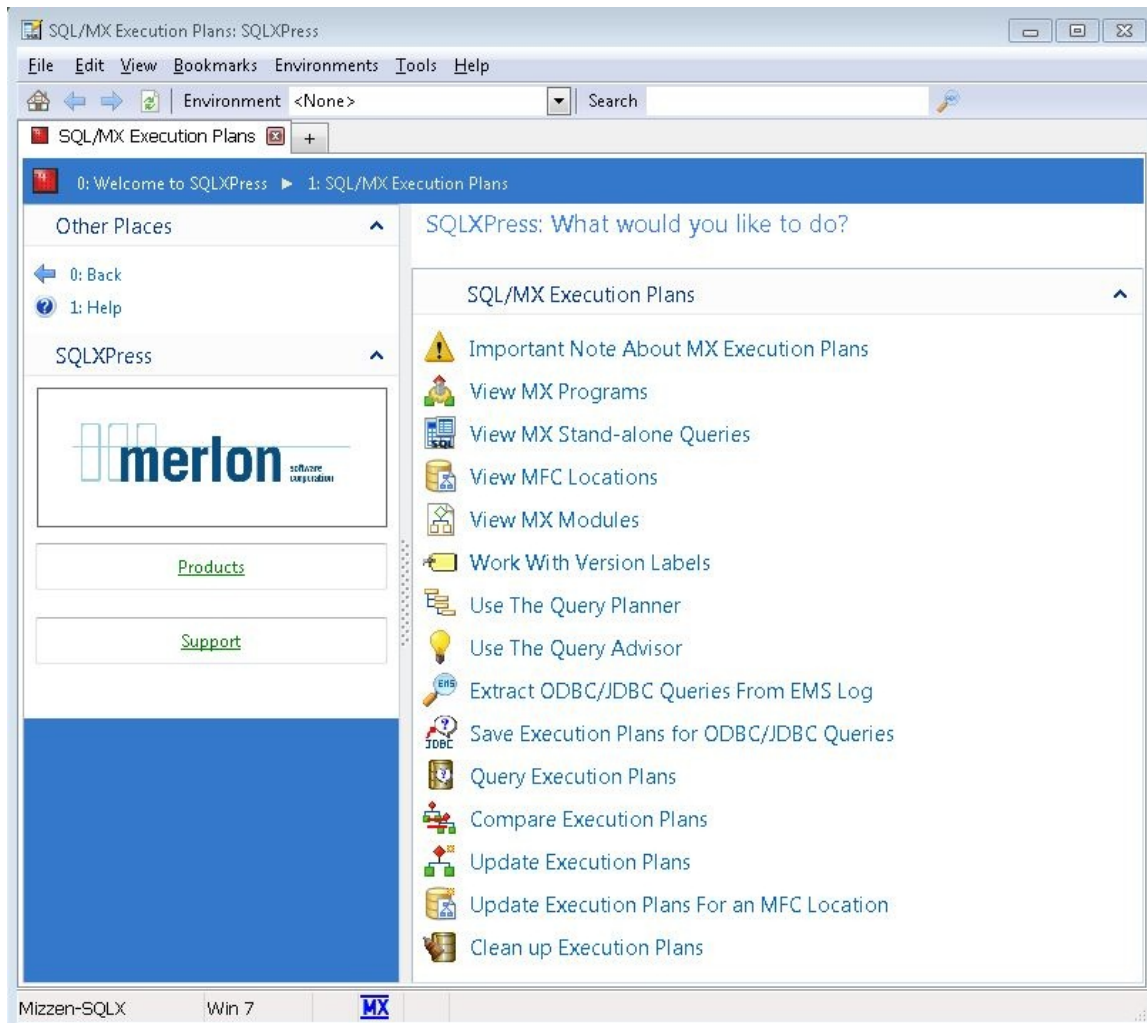
- **Client.** The name of the remote system where the client application is running.
- **Datasource.** The name of the datasource.
- **Server.** The process name of the datasource server.
- **TCP Process.** The name of the TCP process used to communicate with the remote system.
- **Port.** The port number used by the server process to communicate with the remote system.
- **Start Time.** The time at which the session started.
- **End Time.** The time at which the session ended.
- **Selects.** The number of SELECT statements executed during the session.
- **Inserts.** The number of INSERT statements executed during the session.
- **Updates.** The number of UPDATE statements executed during the session.
- **Deletes.** The number of DELETE statements executed during the session.
- **Prepares.** The number of Prepare operations performed during the session.
- **Executes.** The number of Execute operations performed during the session.
- **Execute Directs.** The number of Execute Direct operations performed during the session.

Module File Cache Management

Once MXCS services and datasources have been configured to use the Module File Cache (MFC), SQLXPress can be used to monitor and manage the Module Files themselves.

Most MFC related functions and reports involve the SQLXPress Execution Plan Database, which is created by SQLXPress when it is installed. SQLXPress can also be used to manage execution plans for SQL/MX and SQL/MP programs and queries – this is discussed in a separate document entitled [Managing Execution Plans with SQLXPress](#), available from the Merlon website.

Module File Cache functions are found in the “Work with Execution Plans” entry of the home window:

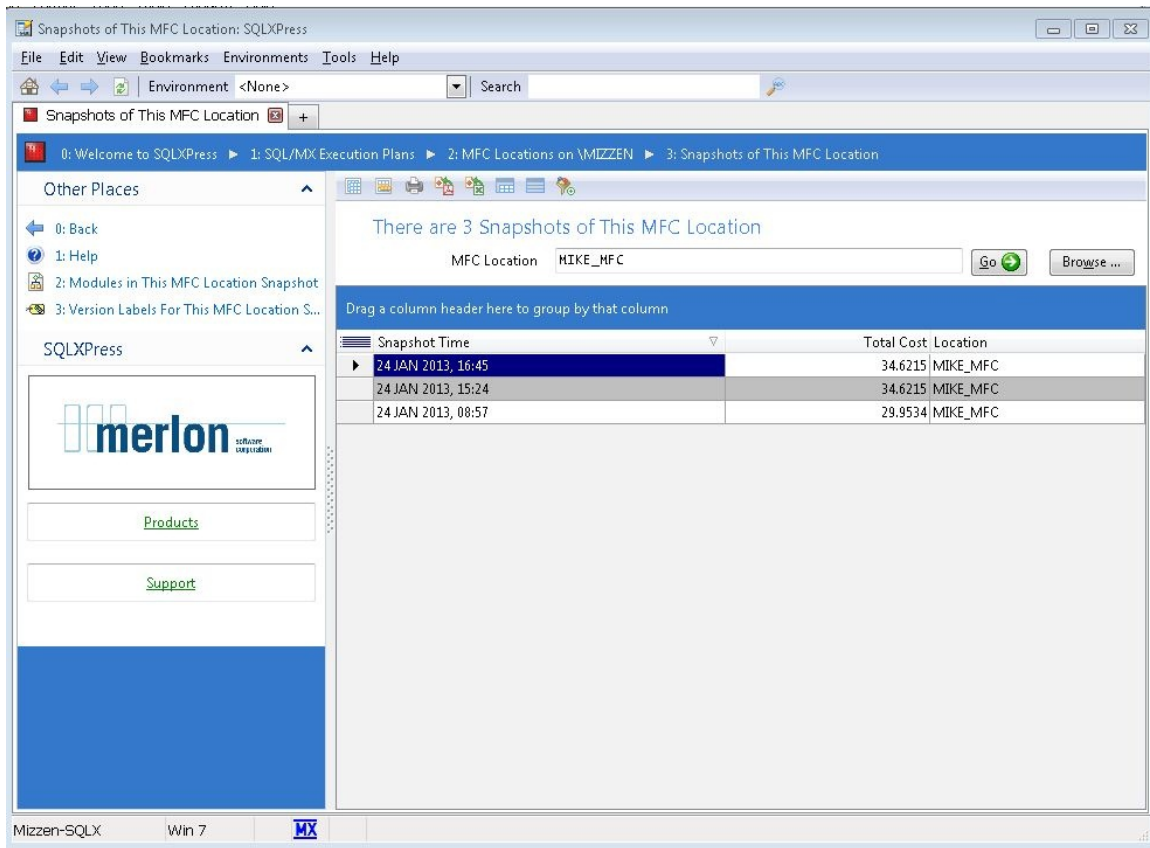


Extracting MFC Execution Plans

To create information in the execution plan database for a MFC location, choose “Update Execution Plans for an MFC Location”. This will start the wizard that allows you to select a physical OSS directory containing the modules, and to assign a logical name for the MFC location. Once you have made the appropriate selections, SQLXPress will run a background job that extracts the plans from the module files and enters them into the SQLXPress database. This process creates a “snapshot”, i.e. an image of the module files and plans at the time you run the wizard. Later, as database or application changes cause modules to be regenerated, SQLXPress can be used to capture additional snapshots. Snapshots can then be compared to identify differences in plans.

Browsing MFC Execution Plans

You can browse snapshots using the View MFC Locations function. Choose a location, and click “Snapshots of this MFC Location”.



The screenshot shows the SQLXPress application interface. The main window is titled "Snapshots of This MFC Location: SQLXPress". The breadcrumb navigation shows the path: 0: Welcome to SQLXPress > 1: SQL/MX Execution Plans > 2: MFC Locations on VMIZZEN > 3: Snapshots of This MFC Location. The left sidebar contains "Other Places" with links for Back, Help, and version labels, and a "SQLXPress" section with the Merlon logo and links for Products and Support. The main content area displays "There are 3 Snapshots of This MFC Location" for the MFC Location "MIKE_MFC". Below this is a table with columns for Snapshot Time, Total Cost, and Location. The table contains three rows of snapshot data.

Snapshot Time	Total Cost	Location
24 JAN 2013, 16:45	34.6215	MIKE_MFC
24 JAN 2013, 15:24	34.6215	MIKE_MFC
24 JAN 2013, 08:57	29.9534	MIKE_MFC

Double click on a particular time to see all of the MFC modules in the snapshot. The modules are presented in order of cost (most expensive first). This gives you an immediate picture of the expensive queries. To see the actual query text, double click on the entry of interest:

Statements in module snapshot /usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.1122D47E9AB9EF8CDE5C334F078382F4 @ 23 JAN 2013, 14:34: SQLXPress

Environment <None> Search

4: Modules in MFC Location snapshot MIKE_MFC @ 24 JAN 2013, 16:45 ▶ 5: Statements in module snapshot /usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.1122D47E9AB9EF8CDE5C334F078382F4 @ 23 JAN 2013, 14:34

Other Places

- 0: Back
- 1: Help
- 2: Execution Plan For This Statement
- 3: Execution Plan Summary For This State...

Related Tasks

- 4: Use The Query Planner

SQLXPress

merlon software corporation

Products

Support

There are 9 Statements in module snapshot /usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.1122D47E9AB9EF8CDE5C334F078382F4 @ 23 JAN 2013, 14:34

Module Snapshot 23 JAN 2013, 14:34 Go Browse ...

Drag a column header here to group by that column

Cost	Cardinality	Verb	Statement Text
29.8792	505	Select	DECLARE MXSTMT01 CURSOR FOR SELECT * FROM MIKE32.TTT.ORDHDR AS O WHERE CUSTOMER_NUM = 3150933
0.0001	1	Other	CONTROL QUERY DEFAULT ODBC_PROCESS 'TRUE'
0.0001	1	Other	CONTROL QUERY DEFAULT JDBC_PROCESS 'TRUE'
0.0001	1	Other	CONTROL QUERY DEFAULT GENERATE_EXPLAIN 'ON'
0.0001	1	Other	CONTROL QUERY DEFAULT NAMETYPE 'ANSI'
0.0001	1	Other	CONTROL QUERY DEFAULT INFER_CHARSET 'ON'
0.0001	1	Other	CONTROL QUERY DEFAULT RECOMPILATION_WARNINGS 'ON'
0.0001	1	Other	DECLARE SCHEMA MIKE32.TTT
0.0001	1	Other	SET SCHEMA MIKE32.TTT

Mizzen-SQLX Win 7 MX

Finding MFC Execution Plans with specific Criteria

Often, there are too many MFC modules to simply browse through the high cost ones. You can locate statements that have particular attributes by using the “Query Execution Plans” function from the “Work with Execution Plans” page.

For example, In the first page, choose “MFC Locations” as the scope of the query. Choose “Most Recent Snapshot” and press continue. On page 2, press the + button in item 2 to add a criterion to the query. You can select a number of different properties to be examined, but in this example choose “Operator” for property, “Equals” for comparison, and “FILE_SCAN” for value. Press continue and click the “Run Query” button. The results are returned, sorted by cost.

Locating Inefficient Plans or Ineffective Plans

As database changes occur, or when SQL/MX statistics are updated, MFC plans can become less optimal, or even completely ineffective: for example, if the MFC plan refers to an index that no longer exists, SQL/MX will find the plan in the Module File Cache, but because it contains an unfulfillable operation, the entire query will be recompiled each time it is first received by a server instance. This can result in poorer performance than if there were no Module File Cache at all. SQLXPress can assist in locating such plans with the “MFC Check” report.

This report is run from the “Compare Execution Plans” function from the “Work with Execution Plans” page. Choose the “Compare the stored and current plans for MFC queries” entry for this report.

Sample output:

```
SQLXPress PCOMP (Version 3.21)
SQLXPress execution-plan database comparison
Copyright (C) 2004-2013, Merlon Software Corporation. All rights reserved
PCOMP command options:
  --checkpath --mfccheck -o /usr/home/mike/TESTMOD
```

```

Validating MFC Module:
"/usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.1122D47E9AB9EF8CDE5C334F078382F4"
  File Modified: 23-Jan-2013, 14:34:42.755956
  Plan Created: 23-Jan-2013, 14:34:42.755956

  >>CONTROL QUERY DEFAULT ODBC_PROCESS           'TRUE'
  >>CONTROL QUERY DEFAULT JDBC_PROCESS           'TRUE'
  >>CONTROL QUERY DEFAULT GENERATE_EXPLAIN       'ON'
  >>CONTROL QUERY DEFAULT NAMETYPE              'ANSI'
  >>CONTROL QUERY DEFAULT INFER_CHARSET         'ON'
  >>CONTROL QUERY DEFAULT RECOMPILATION_WARNINGS 'ON'
  >>SET SCHEMA MIKE32.TTT

Generating plan:
  SELECT * FROM MIKE32.TTT.ORDHDR AS O WHERE CUSTOMER_NUM =
  3150933
!!! Access path 1 (node 1) differs:
"MODULE (MXSTMT01)":
  FILE_SCAN(O (MIKE32.TTT.ORDHDR))
  Begin-key:
    (BUSINESS_UNIT_ID = <min>), (ORDER_NUM_HASH = <min>), (ORDER_NUM = <min>),
    (SYSKEY = <min>)
  Total cost 29.8719
"CURRENT-PLAN (MXSTMT01)":
  INDEX_SCAN(O (MIKE32.TTT.ORDHDR))
  Begin-key:
    (MIKE32.TTT.CUSTX.CUSTOMER_NUM = 3150933), (MIKE32.TTT.CUSTX.BUSINESS_UNIT_ID =
    <min>), (MIKE32.TTT.CUSTX.ORDER_NUM_HASH = <min>), (MIKE32.TTT.CUSTX.ORDER_NUM
    = <min>), (MIKE32.TTT.CUSTX.SYSKEY = <min>)
  Total cost 0.02655
!!! Differences in
"/usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.1122D47E9AB9EF8CDE5C334F078382F4"
-----

Validating MFC Module:
"/usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.2B3170EE75B7032CFAFFC431FAD0B9F8"
  File Modified: 21-Jan-2013, 10:42:18.147095
  Plan Created: 21-Jan-2013, 10:42:18.147095

  >>CONTROL QUERY DEFAULT ODBC_PROCESS           'TRUE'
  >>CONTROL QUERY DEFAULT JDBC_PROCESS           'TRUE'
  >>CONTROL QUERY DEFAULT GENERATE_EXPLAIN       'ON'
  >>CONTROL QUERY DEFAULT NAMETYPE              'ANSI'
  >>CONTROL QUERY DEFAULT INFER_CHARSET         'ON'
  >>CONTROL QUERY DEFAULT RECOMPILATION_WARNINGS 'ON'
  >>SET SCHEMA MIKE32.TTT

Generating plan:
  SELECT * FROM MERLON.SQLEXPRESS.X21PATH WHERE PATHTYPE = 'M'
!!! Access path 1 (node 1) differs:
"MODULE (MXSTMT01)":
  INDEX_SCAN(MERLON.SQLEXPRESS.X21PATH)
  Begin-key:
    (MERLON.SQLEXPRESS.X21HASH.PATHTYPE = 'M'), (MERLON.SQLEXPRESS.X21HASH.PATHHASH
    = <min>), (MERLON.SQLEXPRESS.X21HASH.PATHID = <min>)
  Total cost 0.008729
"CURRENT-PLAN (MXSTMT01)":
  FILE_SCAN(MERLON.SQLEXPRESS.X21PATH)
  Begin-key:
    (PATHID = <min>)
  Total cost 0.033958
!!! Differences in
"/usr/home/mike/TESTMOD/MFCABS1031MIKE32.TTT.2B3170EE75B7032CFAFFC431FAD0B9F8"
-----

Objects examined:      8
Comparisons:          2
Differences:           2

STOPPED: $X5QD

```

Determining the Effect of Changes to the Database or to Control-Query-Defaults

Often, Control Query Default statements (CQDs) can have a significant effect on the plan and therefore the performance of a single query. SQLXPress can assist by comparing plans from an MFC location that have the same statement text, but which reside in different modules. Because the module file name is constructed using a hash of all of the CQDs and the statement text, queries that are executed with different CQDs end up in different MFC module files.

It is also possible to use this report to compare plans generated by different copies of a database that reside in different schemas. The two databases might differ in structure (e.g. there might be indexes in one but not the other), or differ in table size or in the accuracy of their statistics. Such queries need to be coded using a default catalog and schema. The DML statement must not itself reference the catalog and or schema (i.e. the text of the DML statement in both cases is the same, but the default catalog and schema are changed to use different copies of the database). The default catalog and schema contribute to the name of the MFC module file and so two different module files will be created.

Before running this report, ensure that the query or queries of interest have been executed with each set of CQDs or default schemas. You must also update the execution plans for the MFC location.

This report is run from the "Compare Execution Plans" function from the "Work with Execution Plans" page. Choose the "Compare the plans for queries within a single MFC location snapshot" entry for this report.

Sample output:

```
SQLXPress execution-plan database comparison
Copyright(C) 2004-2013, Merlon Software Corporation. All rights reserved
PCOMP command options:
--usemfctext -X -M MIKE_MFC(26 FEB 2013, 11:42) --checkcost --significantcost 0
--tolerance 0 -checkpath

Matched statement:
      DECLARE MXSTMT01 CURSOR FOR SELECT UWORD FROM ENGLISH WHERE
      PARTOFSPEECH = '/a./' AND UWORD BETWEEN 'P' AND 'Q'
(Costs: 5.1343 and 0.092287)
Statement Comparison:
  /usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT.CA72208E27AA7C10A5886D8D40A962BC,
  and /usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT2.CA72208E27AA7C10A5886D8D40A962BC
      DECLARE MXSTMT01 CURSOR FOR SELECT UWORD FROM ENGLISH WHERE
      PARTOFSPEECH = '/a./' AND UWORD BETWEEN 'P' AND 'Q'

!!! Access path 1 (node 1) differs:

/usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT.CA72208E27AA7C10A5886D8D40A962BC (MXSTMT01) ":
  FILE_SCAN (MERLON.DICT.ENGLISH)
  Begin-key:
    (UWORD = 'P'), (DEFNUM = <min>)
  Total cost 5.13361

/usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT2.CA72208E27AA7C10A5886D8D40A962BC (MXSTMT01) ":
  INDEX_SCAN (MERLON.DICT2.ENGLISH)
  Begin-key:
    (MERLON.DICT2.POS_INDEX.PARTOFSPEECH = '/a./'), (MERLON.DICT2.POS_INDEX.UWORD =
    'P'), (MERLON.DICT2.POS_INDEX.DEFNUM = <min>)
  Total cost 0.088336
!!! Plans differ for statement "MXSTMT01"
  Cost for "MXSTMT01" in
  /usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT.CA72208E27AA7C10A5886D8D40A962BC": 5.1343
  Cost for "MXSTMT01" in
  /usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT2.CA72208E27AA7C10A5886D8D40A962BC": 0.092287

8 statements match; 1 statements differ.
!!! "/usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT.CA72208E27AA7C10A5886D8D40A962BC" and
  /usr/home/mike/TESTMOD/MFCABS1031MERLON.DICT2.CA72208E27AA7C10A5886D8D40A962BC" differ.
-----

Objects examined:          5
Comparisons:              1
Differences:              1
```