datamonitor

for iSeries™

Data Monitor for iSeries

User Guide

5.1

DMI-BAS

tango04

Computing Group

Solutions for Advancing People

Data Monitor for iSeries User Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Copyright Notice

Copyright © 2013 Tango/04 All rights reserved.

Document date: July 2012

Document version: 1.1

Product version: 5.1

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Tango/04.

Trademarks

Any references to trademarked product names are owned by their respective companies.

Technical Support

For technical support visit our web site at www.tango04.com.

Tango/04 Computing Group S.L.
Avda. Meridiana 358, 5 A-B
Barcelona, 08027
Spain

Tel: +34 93 274 0051

Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

© 2013 Tango/04 Computing Group

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Appendices

How to Use this Guide

This chapter explains how to use Tango/04 User Guides and understand the typographical conventions used in all Tango/04 documentation.

Typographical Conventions

The following conventional terms, text formats, and symbols are used throughout Tango/04 printed documentation:

Convention	Description
Boldface	Commands, on-screen buttons and menu options.
Blue Italic	References and links to other sections in the manual or further documentation containing relevant information.
Italic	Text displayed on screen, or variables where the user must substitute their own details.
`Monospace`	Input commands such as System i commands or code, or text that users must type in.
UPPERCASE	Keyboard keys, such as CTRL for the Control key and F5 for the function key that is labeled F5.
✎	**Notes** and useful additional information.
💡	**Tips** and hints that will improve the users experience of working with this product.
⚠	**Important** additional information that the user is strongly advised to note.
❗	**Warning** information. Failure to take note of this information could potentially lead to serious problems.

Chapter 1
Introduction

Data Monitor as the ultimate auditing tool

Data Monitor is an iSeries auditing tool that efficiently collects data from journals and monitors data at field-level, is based on dynamic policies, and is capable of monitoring data in real-time and in multiple database files simultaneously. With Data Monitor you can monitor for example unexpected access, malicious modifications and changes that compromise data integrity.

> *"Data Monitor for iSeriesis incredibly easy to use. Not only does it help keep the auditors at bay, but it's also nice from an internal standpoint to help track when a user creates a data issue, be it accidental or otherwise."*
>
> *Rated 3.7 out of 4 (equivalent to 9 out of 10).*
>
> *Taking Some Pain Out of IT Auditing, IBM Systems Magazine*

Use Data Monitor to audit your iSeries and make sure you comply with laws regarding data access and protection, such as Sarbanes-Oxley (SOX), and Payment Card Industry (PCI) Data Security Standard. Armed with Data Monitor you can evaluate your company's compliance with practically any kind of regulation, or corporate standards, both for technical staff and auditors with little knowledge of the underlying it infrastructure.

Data Monitor for iSeries Base uses OS native journaling only[1] and supports OS/400 V5R1 and later versions. It lists record-level transactions such as inserts, deletes, reads and updates, as well as the DLTF, CLRPFM and ENDJRNPF commands, in extremely easy to read, graphical reports. Once Data Monitor processes transactions from a journal, the journal receivers are no longer required for Data Monitor to work. The receivers can be saved for legal purposes and to recreate transaction data from journal receivers at any time.

Data Monitor generates a series of internal transaction files, known as the Historical Transaction Log. Use VISUAL Message Center Reports to generate several kinds of reports based on data stored in the Historical Transaction Log.

Get the most out of your Data Monitor by using it in combination with VISUAL Message Center SmartConsole and VISUAL Message Center Reports. Together with these Tango/04 products Data Monitor becomes an efficient auditing and tracing tool, that allows for increased security of data storage, historical analysis, fraud detection, or forensic analysis, and can be used as an early warning system for prevention and quick response, attended or not, to attempted intrusions by hackers.

[1] The read-only access auditing uses database triggers. Triggers are not needed for auditing of changes.

Data Monitor as a Real time protection tool

Data Monitor is unique in the sense that it can generate real time alerts for practically any kind of suspicious activity. For instance, too high discounts, movements in dormant accounts, late night changes, application bypassing (by using SQL or programming tools), privileged user access, and more. These alerts can be sent to the VISUAL Message Center SmartConsole and escalated using cellular phones or email, and even automated action (such as blocking a user account) can be performed without human intervention.

Data Monitor is specially rich in the kind of auditing controls that can be set on data, since it has a powerful scripting language, generates extensive metadata on changes (such as user group, class, accounting code, and so on), and provides for easy ways to access data from other databases, effectively controlling all user activity and proactively mitigating risk.

Data Monitor as a KPI (Key Performance Indicator) monitoring tool

Data Monitor for iSeries can be used for more than just auditing. When dealing with high-volume, highly interactive industries such as banking, retail, finance, manufacturing, or insurance, there is a need to control certain performance indicators that are extremely costly to retrieve using standard SQL statements. Data Monitor can help here, too, by reacting immediately to changes instead of periodically running resource-hogging query statements to check data. Real time control of several KPIs such as critical stock levels, cash on hand, sales per hour, call center average delay, and more, can be efficiently and easily achieved by using the advanced alerting features of Data Monitor. Alerts of any deviation from normal can be sent to an escalation list using cellular phone messaging or plain email.

Note

Data Monitor does not journal physical files, nor does it create Journals or Journal Receivers. Physical files must be journaled previously using the `STRJRNPF` command, with parameter IMAGES set to (`*BOTH`).

Tip

To maximize the information collected by Data monitor – in versions V5R2 and up – it is important to specify the values `*JOB`, `*USR`, `*PGM`, `*PGMLIB` and `*RMTADR` in the parameter `FIXLENDTA` when creating or changing your journals (`CRTJRN` and `CHGJRN` commands).

1.1 Data Monitor Main Features

Detailed information for each transaction

Data Monitor for iSeries can retrieve detailed information of every change or read, including timestamp, job, user, real user (which may differ from the job user), user class, accounting code, IP address of the remote job that executes the transaction (where available), name of the program and library (for inserts, deletes, and updates only), and more.

User and File monitoring

Data Monitor for iSeries can be set to monitor a group of sensitive files (regardless the user who access them), or a group of sensitive users (regardless of the files they access), or a combination of both.

Extremely low resource consumption

Leveraging an elegant architecture and advanced programming techniques, Data Monitor has usually a negligible impact on performance or even no impact at all (when using the Remote Journal support)[2] for changes auditing[3].

Highly customizable Collection Engine

Data Monitor for iSeries can be customized to monitor only the fields and transactions that really matter, avoiding the storage of unnecessary data. Reports can be customized, indicating which fields are to be listed. Virtual fields can be created, based on existing fields, data in other tables, and formulas.

The reports can extract information at field level. Concentrate on the most critical events and only audit sensitive fields and records and filter out less relevant information at the source.

Flexible reports in real time or on demand

Data Monitor for iSeries uses VISUAL Message Center Reports to generate reports instantly and interactively, or schedule their creation at a later point in time. Reports allows for on-screen display of reports, or in other formats, including:

- Acrobat (PDF)
- Comma Separated Values (CSV)
- Excel (XLS)
- dBase IV (DBF)
- Word (DOC)
- XML

Reports can be sent automatically by E-mail, or feed a corporate Web portal or a restricted area on the Intranet.

Efficient use of storage

As reports are generated from an optimized database, the journal receivers can be erased at any time to free valuable disk space. Moreover, the Historical Transaction Log can be stored on a remote database and even on a Windows, Linux or UNIX server.

Note
Data Monitor for iSeries Express requires the journal receivers to be online.

Automatic record format discovery and refresh

Data Monitor for iSeries adapts to application changes. The fields are always displayed correctly regardless of format changes, removal or addition of fields, and other modifications.

If desired, it takes care of loading formats for files without user intervention initially. For more details *see Chapter 4 - Configuring and Starting Data Monitor to monitor Sensitive Users on page 43.*

It also automatically refreshes record formats as they are changed. For details *see Chapter 6 - Handling formats in Data Monitor for iSeries on page 58.*

Customized sub-reports

Each authorized user can easily create customized sub reports based on predefined reports included in Reports. For example, they can insert logos, modify the appearance of lists, and use an intuitive graphical interface to create combinations of parameters and quickly respond to questions like:

[2] There will be no impact on performance on the Production machine since the transaction processing will be done entirely on the remote (usually a backup) machine. This feature can coexist with popular High Availability products, such as VISION Solution ORION, iTera, MiMiX, or DataMirror HA Suite or iCluster.

[3] Read-only access may have performance impact and should be deployed with care. See the chapter on Performance Considerations for more information.

- Who made changes to the file `PAYROLL` accessing the file as a `*SECOFR`-class user?
- Who modified `INVOICES` between 00:00 and 05:00?
- What changes (to any table) did user `JONES` make in the last month?
- What were all the changes made to the sensitive field `DISCOUNTS` in the last quarter?
- What changes were made to `CUSTOMERS` not using the program `FINANCE/ACTCLI04`?
- What changes were made via `DFU`?
- How many changes were made to the table `PAYCHECK` in the last year?
- What users that do not belong to the `HR` user group modified the table `SALARIES`?

Before and after image

If desired, the before and after values can be shown for every changed field. It is also possible to define the order in which the fields appear or to exclude one or more fields altogether.

Support with legal validity

When uncovering fraudulent behavior by an employee, or any other suspect transaction that requires contacting national or foreign justice, Data Monitor for iSeries can help. It is based on data on the IBM journal receivers, which have been certified by IBM as “usable as legal proof” in the United States and other countries. This is not the case with other methods of capturing data, such as triggers, used by other products in the market today.

Quick to start and easy to use

Data Monitor for iSeries was designed to be put into production with minimal effort on the user's part and with a very low operating cost.

Time-saving Graphical, Comprehensive Reporting

Data Monitor graphical Reports are not only eye pleasing, but also extremely powerful for rapidly spotting problem areas. Several features contribute to save time: clearly marked field changes, complete transaction metadata, the ability to retrieve Key Descriptions (showing, for instance, customer name and surname in addition to customer code), the ability to create virtual fields with extended information from other tables (such as customer type, customer address, sales rep name, or any other data from master or secondary files), the ability to easily generate reports with several parameters (date, user, program, etc.), the ability to report changes to a certain field only, the ability to report changes where certain field assumed certain value, and the ability to be automatically alerted to any suspicious condition in real time, and more. All this power saves you a lot of time and ensures that all potentially risky situations are handled properly.

Transparent to existing applications

As it directly integrates with the database journal receivers, Data Monitor for iSeries does not require any changes to be made to existing applications.

Complete integration with VISUAL Message Center

Data Monitor for iSeries seamlessly integrates with VISUAL Message Center and VISUAL Security Suite, thereby taking full advantage of the power of the SmartConsole in generating early warnings, message escalation and automation of actions.

Protection of confidential information

The most sensitive data can be monitored openly or *obscured*. For example, it is possible to hide data such as account balance, credit card numbers, phone numbers, and more. This data will appear in reports as `***Restricted***`.

> ⚠ **Important**
>
> Data Monitor for iSeries provides all the tools needed to ensure high level sensitive data is correctly protected according to European Data Protection Directives.
>
> For further information please refer to *see Appendix A: Recommendations for Protecting High Level Data on page 93*.

Remote Journal Support

Data Monitor for iSeries can be installed on a backup machine and collect data from a remote journal, avoiding the usage of any resources on the production server.

Resiliency, support for reprocessing and cross checking

In the event that you need to change the custom format of a table, or if you need to modify the conditions to audit a transaction, you can easily reprocess a group of transactions, indicating the starting date to reprocess from. Transactions are never repeated to ensure database integrity. Cross-checking of the database with the original journal is available if required, and automated resiliency processes guarantee that no transaction is lost even if the monitor is deactivated for a while. Any attempt to remove the data from journaling is promptly detected and reported, and an alert can be generated in real time.

Self-managing database

Data Monitor for iSeries includes self-cleaning commands that can be easily scheduled to achieve self-management of the historical database.

Automatic Operations

Operation of Data Monitor for iSeries (Base) can be completely automated. Daily starting and ending is automatic, the database formats are refreshed automatically when a table records format changes, and combined with the automated alerts and automated creation of reports, fully automated operations can be easily achieved.

Open architecture for unlimited application integration

Data Monitor for iSeries can call an external RPG, C, Java or COBOL program (or any other i5/OS supported program) to execute an action or to calculate the value of a field. For instance, if you have a complex and fully tested RPG program that calculates interest rates for an account, and you want to include the results of that calculation in an auditing filter, alert rule, or a report, you can call the RPG program using the so-called External Business Function, which basically calls any program and passes back calculated data.

Support for latest IBM technologies

Data Monitor for iSeries continuously adds the latest IBM enhancements to the journaling and operating systems. For instance, it supports all the information on the newest TYPE5 journal entries, adding important information such as IP Address and Program Library. The most advanced programming techniques available in each i5/OS release are used. Tango/04 has a direct liaison with the IBM Laboratories in Rochester, Minnesota (USA).

Field proven

Data Monitor for iSeries architecture has been tested in production environments with customers auditing more than one billion transactions per month, with excellent results, low resource consumption, and high customer satisfaction. Customer references are available upon request.

1.2 Benefits of Data Monitor

- Prevent fraud
- Fulfill internal and external auditing requirements
- Maintain a record of changes made to any field of an existing application, without modifying the application
- Comply with specific Sarbanes-Oxley (SOX) requirements and other international or industry regulations (such as Basel II for financial companies, 21 CFR Part 11 for the food and pharmaceutical industry, HIPAA for healthcare companies, PCI and so on)
- Create advanced reports for rapid detection and resolution of inappropriate or illegal use
- Efficiently store changes to sensitive or critical data on secure systems (different from the original system if desired)
- Meet international security standards such as ISO 17799, adapt to security management recommendations by ITIL, CobiT and other industry best practices
- Count on a record of individual user activity to prove illegal activities
- Dissuade employees and temporary hires from entering into improper activities by implementing an advanced data monitoring system
- Real-time, efficient monitoring of KPIs (Key Performance Indicators).

1.3 Understanding Data Monitor Versions

Data Monitor for iSeries has a standard (Base) version that includes most features of the product, with the exceptions mentioned below. One distinctive characteristic of this version is that it does not need the journal receivers of the monitored files to be stored online, as the transactions are stored on a separate database (the Historical Transaction Log). This makes it very easy to get a report on user activity at any time (including the inspection of activity that happened a long time ago) regardless whether the original journal is still on disk.

Data Monitor for iSeries Express is a subset of the base product, which can only retrieve data from a Journal receiver that is present online. If you need to obtain a report on past activities using the Express version and the Journal receiver does not longer contain the required data, you will need to restore the old journal receivers with the data first.

It is possible to upgrade from the Express to the Base version. Contact your Tango/04 Authorized Partner to know more about the upgrade process.

There is a separate license for monitoring read-only access to files. If you need to monitor and audit read only access, contact your Tango/04 Authorized Partner.

Chapter 2
Installing Data Monitor for iSeries

The Data Monitor for iSeries is installed as a stand-alone product. It generates data on the server, which is then used by VISUAL Message Center Reports to generate a series of auditing reports.

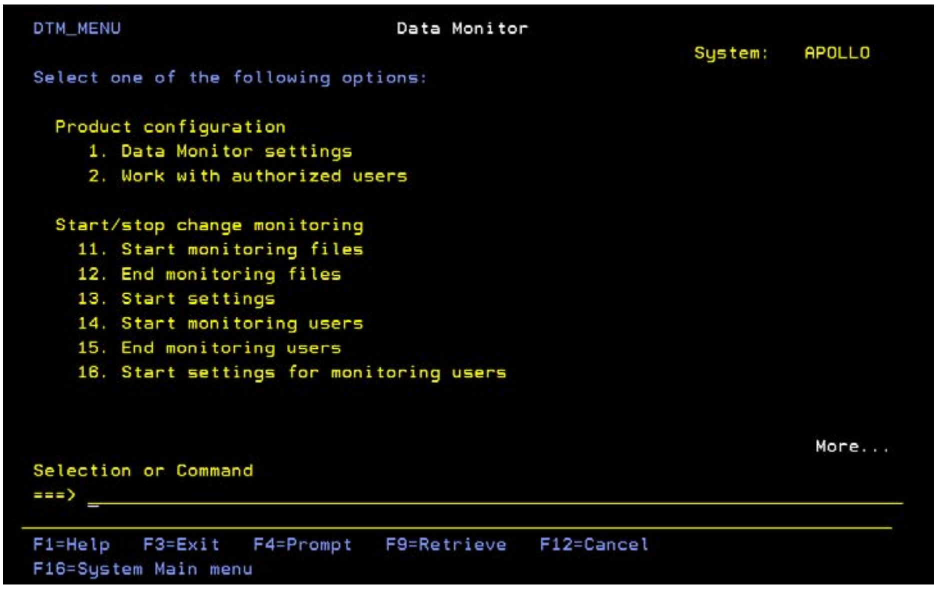
To get the most out of the Data Monitor for iSeries you should install both Data Monitor for iSeries Data Monitor for iSeries and VISUAL Message Center Reports. For more information *see Chapter 10 - Data Monitor Reports on page 79*.

Data Monitor can be installed while other applications are using the files, journals and journal receivers.

2.1 Data Monitor Menus

After you have installed the product in the `T4DATAMON` library, you can access the main menu using the following command:

```
GO T4DATAMON/DTM_MENU
```



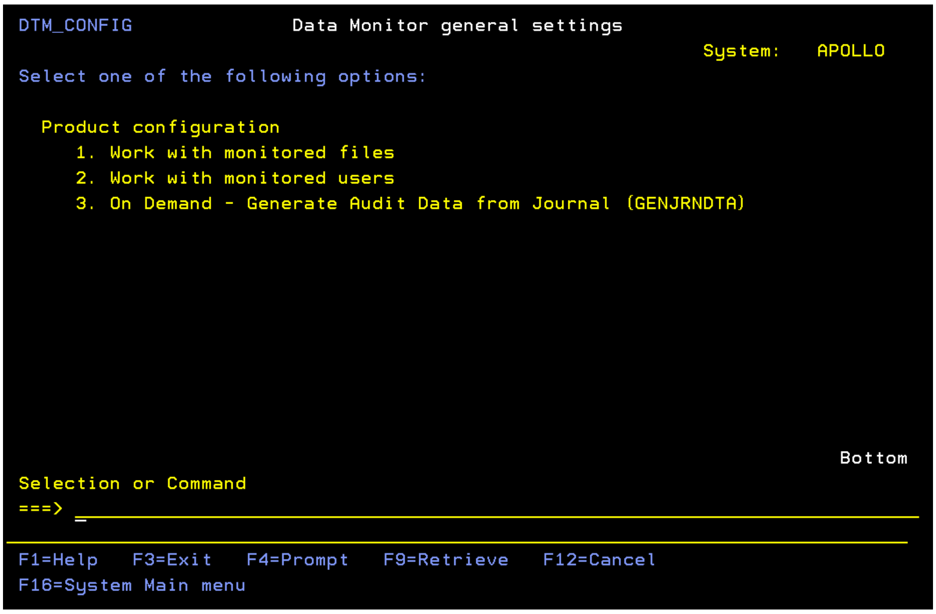
```
DTM_MENU                         Data Monitor
                                                        System:   APOLLO
Select one of the following options:

  Product configuration
     1. Data Monitor settings
     2. Work with authorized users

  Start/stop change monitoring
    11. Start monitoring files
    12. End monitoring files
    13. Start settings
    14. Start monitoring users
    15. End monitoring users
    16. Start settings for monitoring users


                                                                  More...
Selection or Command
===> _

F1=Help   F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F16=System Main menu
```

Figure 1 – Data Monitor main menu

Menu `DTM_MENU` displays the screen as shown in *Figure 1* above and contains the main options for working with Data Monitor. Press PgDn for more menu options.

To open the configuration of Data Monitor select option **1** Data Monitor settings. The General Settings menu will appear, where you can add the sensitive files or sensitive users you want to monitor and work with different formats. Another way to open the General Settings menu is to enter the following command:

```
GO T4DATAMON/DTM_CONFIG
```



```
DTM_CONFIG                Data Monitor general settings
                                                          System:    APOLLO
Select one of the following options:

  Product configuration
     1. Work with monitored files
     2. Work with monitored users
     3. On Demand - Generate Audit Data from Journal (GENJRNDTA)









                                                                    Bottom
Selection or Command
===> _

F1=Help   F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F16=System Main menu
```

Figure 2 – Option 1: Data Monitor General Settings menu (menu DTM_CONFIG)

See the next chapter for details regarding configuration.

2.2 Reports

In VISUAL Message Center Reports you will find a number of audit reports designed specifically for use with Data Monitor.

Reports uses an ODBC connection to retrieve data from Data Monitor's database. The Data Monitor database located on an iSeries system can also be replicated to different platforms such as Oracle or SQL Server. Reports can use any of these databases for its reports.

For details regarding the installation of Reports and ODBC see the *VISUAL Message Center Reports User Guide*.

2.3 Product Activation Keys

Data Monitor for iSeries can be used to monitor a variety of files and users. You can choose to monitor only changes, or also monitor reads. Alternatively you can choose to use Data Monitor for iSeries Express simply as a basis for reporting. Separate keys are available depending on how you intend to use the product.

2.3.1 Changes and Reads

There are separate Product Activation Keys for monitoring changes (insert, delete and modify) and for monitoring reads, even though it is possible to process reads and changes in the same monitor, both for sensitive files and for sensitive users.

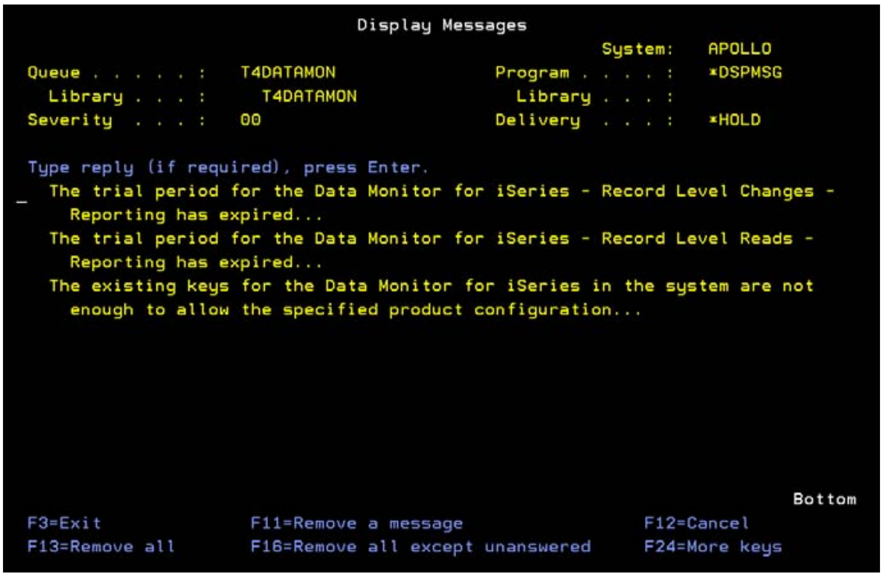
If you only have a key to monitor changes and you configure a monitor to also check reads of a monitored file, the monitors will not start. A message appears informing you that you do not have a valid read key. Of course as long as you configure monitors only for checking changes, the monitor will start.

Likewise, if you only have a key to monitor reads and you configure a monitor to also check changes of a monitored file, the monitor for this monitored file will not start.

For example: Let's say you have a product key for changes only and there are three monitored files on your system. Monitored file A monitors both changes and reads. Monitored file (B) monitors changes only and monitored file C only monitors reads.

When you start your monitors only the monitor for file B is active. The monitor for file A is not activated because not all the required keys are present.

Contact your Tango/04 partner for Product Activation Keys.



```
                                 Display Messages
                                                      System:   APOLLO
 Queue . . . . . :   T4DATAMON                Program . . . . :   *DSPMSG
   Library . . . :     T4DATAMON                Library . . . :
 Severity  . . . :   00                       Delivery  . . . :   *HOLD

 Type reply (if required), press Enter.
_  The trial period for the Data Monitor for iSeries - Record Level Changes -
     Reporting has expired...
   The trial period for the Data Monitor for iSeries - Record Level Reads -
     Reporting has expired...
   The existing keys for the Data Monitor for iSeries in the system are not
     enough to allow the specified product configuration...










                                                                       Bottom
 F3=Exit              F11=Remove a message                  F12=Cancel
 F13=Remove all       F16=Remove all except unanswered      F24=More keys
```

Figure 3 – Example of messages for missing keys

2.3.2 Data Monitor for iSeries Express

Data Monitor for Express offers you a snapshot of existing journal data for reporting purposes. When you are ready to collect journal data, simply enter the required parameters and run the command `GENJRNDTA`. Data Monitor Express allows you to store the collected information (Historical Transaction Log) in a different library. In your reports simply specify the library from which to draw up the report.

Data Monitor for iSeries Express is available as a stand alone product, as part of Data Monitor for iSeries Base, or can be activated by a separate suite key.

Chapter 3

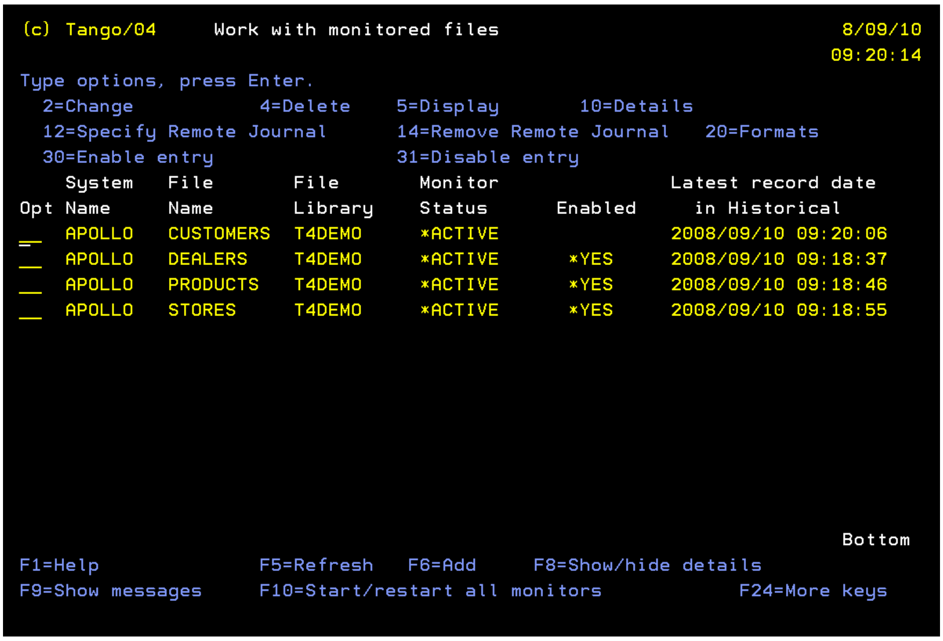
Configuring and Starting Data Monitor to monitor Sensitive Files

Data Monitor for iSeries can be configured to monitor sensitive files and sensitive users. This chapter describes how to set up Data monitor for iSeries to monitor sensitive files. *Chapter 4 - Configuring and Starting Data Monitor to monitor Sensitive Users on page 43* describes how to set Data Monitor for monitoring sensitive users.

The first sections discuss the screens you will use to check your Data Monitor configuration and the basic variables displayed in these screens. The remaining sections discuss how to add new entries and modify existing configuration entries for monitoring sensitive files.

3.1 Viewing Data Monitor Configuration

From the General Settings screen (menu `FILE_MENU`), use option **1** (Work with monitored files) to see all the configuration entries for the Data Monitor.



```
 (c) Tango/04     Work with monitored files                         8/09/10
                                                                   09:20:14
Type options, press Enter.
  2=Change           4=Delete    5=Display       10=Details
  12=Specify Remote Journal      14=Remove Remote Journal   20=Formats
  30=Enable entry                31=Disable entry
    System   File       File       Monitor                Latest record date
Opt Name     Name       Library    Status      Enabled     in Historical
__  APOLLO   CUSTOMERS  T4DEMO     *ACTIVE                2008/09/10 09:20:06
__  APOLLO   DEALERS    T4DEMO     *ACTIVE      *YES      2008/09/10 09:18:37
__  APOLLO   PRODUCTS   T4DEMO     *ACTIVE      *YES      2008/09/10 09:18:46
__  APOLLO   STORES     T4DEMO     *ACTIVE      *YES      2008/09/10 09:18:55









                                                                     Bottom
F1=Help             F5=Refresh    F6=Add     F8=Show/hide details
F9=Show messages    F10=Start/restart all monitors             F24=More keys
```

Figure 4 – Sensitive files to monitor

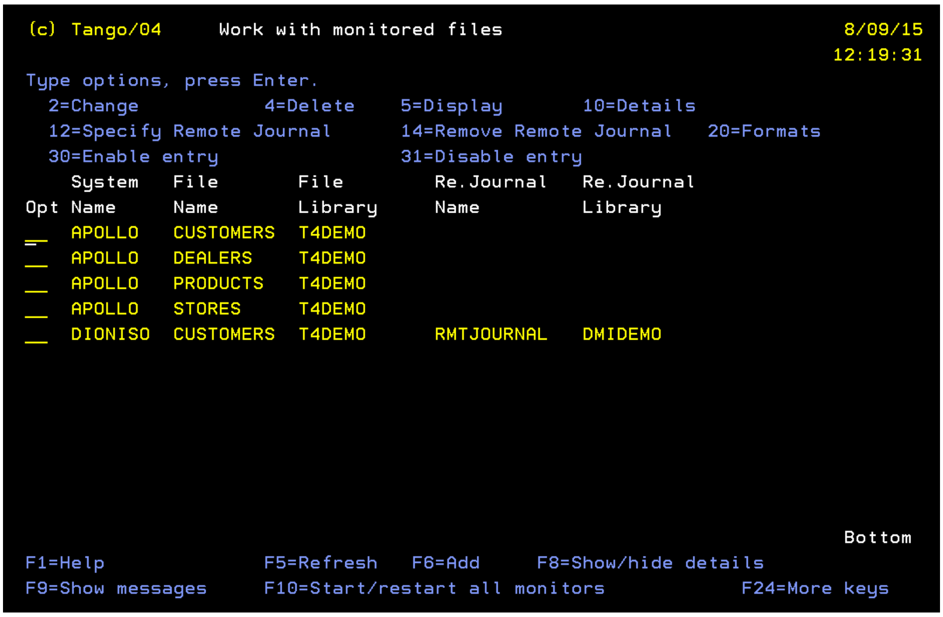
The first three columns (System Name, File Name and File Library) identify the sensitive files that have been added to the Data Monitor configuration.

Monitor Status indicates the job status of the job dedicated to monitoring the configured file. For the Data Monitor to monitor a particular entry the entry must be enabled. An entry that is not enabled will not be monitored when the Data Monitor is restarted. For more information *see section 3.6 - Starting/Restarting Monitors on page 30*.

Latest record date in Historical indicates the date and time of the last record stored in the historical transactions database for the file in this configuration entry.

Of course, if this is the first time you are using the product this screen will be empty. You can add new configuration entries using F6. For details *see section 3.3 - Adding a File to Data Monitor on page 14*

A second view is available by pressing F13. This view shows the remote journal details for each file.



```
(c) Tango/04     Work with monitored files                               8/09/15
                                                                        12:19:31
Type options, press Enter.
  2=Change           4=Delete     5=Display       10=Details
  12=Specify Remote Journal       14=Remove Remote Journal    20=Formats
  30=Enable entry                 31=Disable entry
    System   File       File        Re.Journal   Re.Journal
Opt Name     Name       Library     Name         Library
__  APOLLO   CUSTOMERS  T4DEMO
__  APOLLO   DEALERS    T4DEMO
__  APOLLO   PRODUCTS   T4DEMO
__  APOLLO   STORES     T4DEMO
__  DIONISO  CUSTOMERS  T4DEMO      RMTJOURNAL   DMIDEMO







                                                                       Bottom
F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors           F24=More keys
```

Figure 5 – Alternative view using F13 displays remote journal details for each file.

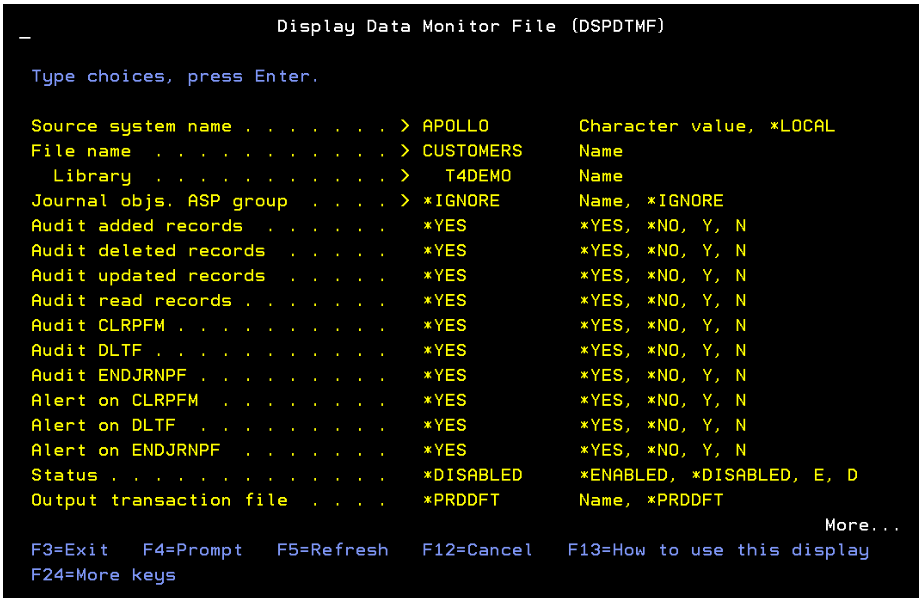
Use options **12** – Specify Remote journal and **14** – Remove Remote Journal to add and delete a configuration entry for remote journals. For details *see section 3.8 - Monitoring Files on Remote Systems on page 35*.

3.2 Displaying File Details in Data Monitor

From the Work with monitored files screen use option **10** to display the details of a monitored file. Another way to display the file details is to enter the following command:

`T4DATAMON/DSPDOMF` and press F4.

Here enter the file you want to view and press F9.



```
_                          Display Data Monitor File (DSPDTMF)

 Type choices, press Enter.

 Source system name . . . . . . . > APOLLO        Character value, *LOCAL
 File name  . . . . . . . . . . . > CUSTOMERS     Name
   Library  . . . . . . . . . . . >   T4DEMO      Name
 Journal objs. ASP group  . . . . > *IGNORE       Name, *IGNORE
 Audit added records  . . . . . .   *YES          *YES, *NO, Y, N
 Audit deleted records  . . . . .   *YES          *YES, *NO, Y, N
 Audit updated records  . . . . .   *YES          *YES, *NO, Y, N
 Audit read records . . . . . . .   *YES          *YES, *NO, Y, N
 Audit CLRPFM . . . . . . . . . .   *YES          *YES, *NO, Y, N
 Audit DLTF . . . . . . . . . . .   *YES          *YES, *NO, Y, N
 Audit ENDJRNPF . . . . . . . . .   *YES          *YES, *NO, Y, N
 Alert on CLRPFM  . . . . . . . .   *YES          *YES, *NO, Y, N
 Alert on DLTF  . . . . . . . . .   *YES          *YES, *NO, Y, N
 Alert on ENDJRNPF  . . . . . . .   *YES          *YES, *NO, Y, N
 Status . . . . . . . . . . . . .   *DISABLED     *ENABLED, *DISABLED, E, D
 Output transaction file  . . . .   *PRDDFT       Name, *PRDDFT
                                                                      More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 6 – Display file details

The file details screen shows the current data for this configuration entry. Note that there are several screens of file details. Press PgDn to see more file details.

Here follows a brief description of the fields available on the first screen of file details.

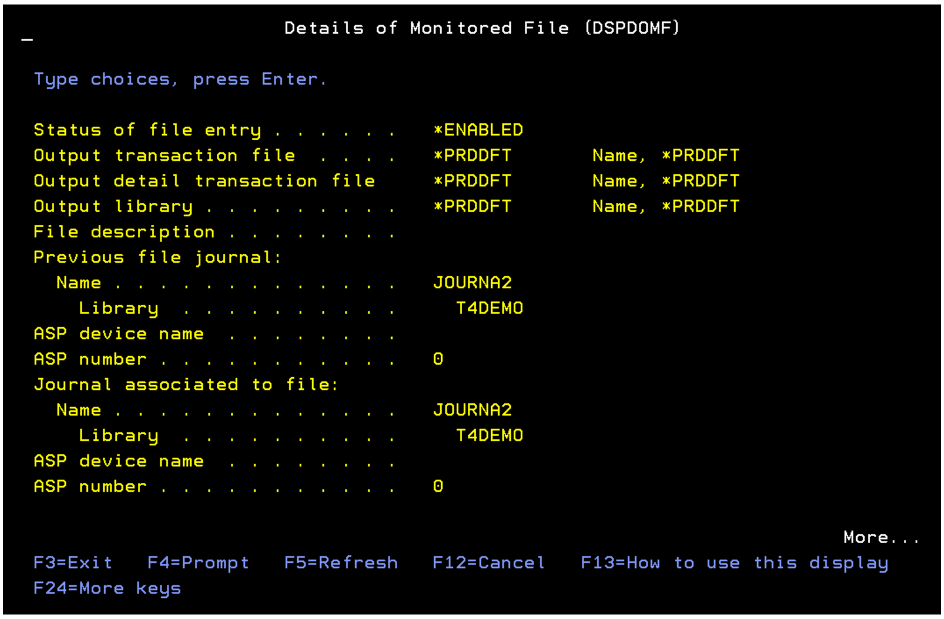
Source system name, Data file name and **Library** identify the sensitive file that is being monitored.

Journal Obj. ASP group (only for V5R2M0 and later): This value is used when searching journal and journal receiver objects where file is journaled. If you were not aware of ASP groups, or do not work with ASP groups in your system, ignore this parameter and set it to the default special value `*IGNORE`. If you work with ASP groups enter the ASP group name here.

File auditing for: indicates what types of transactions or commands will be monitored. If you enter `*YES` for any of these options, they will be available in the Historical Transaction Log and can be used in Reports.

Alerts of file for: Alerts are important events that affect monitoring sensitive files. For example an alert can be sent to VISUAL Message Center when someone ends journaling for a file. Indicate here what alerts will be monitored by VISUAL Message Center. Note that the alerts configuration is independent of the file auditing configuration above.

Press PgDn to display more file details, including:



```
_                         Details of Monitored File (DSPDOMF)

 Type choices, press Enter.

 Status of file entry . . . . . .   *ENABLED
 Output transaction file  . . . .   *PRDDFT       Name, *PRDDFT
 Output detail transaction file     *PRDDFT       Name, *PRDDFT
 Output library . . . . . . . . .   *PRDDFT       Name, *PRDDFT
 File description . . . . . . . .
 Previous file journal:
   Name . . . . . . . . . . . . .   JOURNA2
     Library  . . . . . . . . . .     T4DEMO
 ASP device name  . . . . . . . .
 ASP number . . . . . . . . . . .   0
 Journal associated to file:
   Name . . . . . . . . . . . . .   JOURNA2
     Library  . . . . . . . . . .     T4DEMO
 ASP device name  . . . . . . . .
 ASP number . . . . . . . . . . .   0

                                                                     More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 7 – Display file details, page 2

Status of file entry: the user can enable or disable the configuration entry. The only valid values in this field are `*ENABLED` or `*DISABLED`. When the user executes function F10 (start and reinitiate all monitors) only the enabled entries will be started or reinitialized.

Output transaction file: The name of the physical file used to store the transaction list retrieved by Data Monitor. Currently you can only use the value `*PRDDFT`. Using this value the user indicates that the name of the historical transactions database is `CHLOC01P`.

Output detail transaction file: The name of the physical file used to store the list of transaction details retrieved by Data Monitor. Currently you can only use `*PRDDFT`. Using this value the user indicates that the name of the detailed historical transaction database is `CHDOC01P`.

Output Library: Use this parameter to configure the library name where the physical files or databases of transaction lists retrieved by Data Monitor are stored. Currently you can only select `*PRDDFT`. Using this value the user indicates that the library name will be `T4DATAMON`.

Previous file journal details: name and library of the previous journal registered to the current file, name of the ASP device (version V5R2M0 or later) and the ASP number of the previous journal related to the file that is being viewed.

Current journal details: name and library of the journal currently related to the file, name of the ASP device (versions V5R2M0 or later), and ASP number of the journal related to the file being viewed.

Journal receiver details: name and library of the current journal receiver, name of the ASP device (version V5R2M0 and later) and ASP number of the current journal receiver related to the file that is al being viewed.

Job monitor details: name, user, number and job status of the job currently dedicated to monitoring the file. This job may be simultaneously monitoring other files configured in Data Monitor.

Monitor status: Current monitor status. It indicates the job status of the job dedicated to monitor the file configured in Data Monitor. The following valid values are possible:

Status	Description
*ACTIVE	The monitor is active and in a correct state.
*JOBQ	The monitor is in the queue but has not yet been started. This value is platform-dependent. iSeries uses queues when starting processes.
*OUTQ	The monitor is not active. The previous session ended correctly but left a log.
*INITFAIL	An error occurred while the monitor was in *STARTING status.
*STARTING	The monitor is still in the initialization phase.
*INACTIVE	This monitor is not started and there is no evidence that this monitor has been started in the past.
*ENDING	The monitor is being ended from the interface.
*INERROR	The monitor is active, but not responding.
*FAILED	The monitor ended due to an error that occurred while in *ACTIVE state.
*RESTART	The monitor is being restarted from the interface.
*JRNINACTI	The journal is temporarily inactive. The monitor is active and waiting for the journal to be activated so that it can return to analyzing the operations.
N/A	An error occurred while trying to retrieve the monitor status.

3.3 Adding a File to Data Monitor

Now that you know where to find information about the current configuration we will discuss how to create new entries and modify existing entries.

Although Data Monitor reads transactions from journals, it works at file level, not at journal level. So, if the same journal is used for several journaled files, Data Monitor only processes transactions for the sensitive files that are specified, and not those of the entire journal.

To further narrow down the information monitored by Data Monitor you can specify the type of transaction to process. For example, if you are only interested in the modifications, you can set the inserts, reads and deletes to `*NO` and only the update transactions will be processed. In addition, you can refine the monitoring process down to field level. By defining sensitive fields only transactions related to these fields will be listed.

Note

Data Monitor does not journal physical files. Files must be journaled in advance using the `STRJRNPF` command, with parameter `IMAGES` set to (`*BOTH`). Data Monitor checks this requirement when starting the monitor.

`MINENTDTA(*FILE)` is not supported in Journals. To check the value of parameter `MINENTDTA` use command `WRKJRNA` on the journal.

The journal for the physical file needs to have parameter `MINENTDTA` (minimized entry-specific data) set to a value different from `*FILE`. IBM iSeries Information Center explains this note as follows:

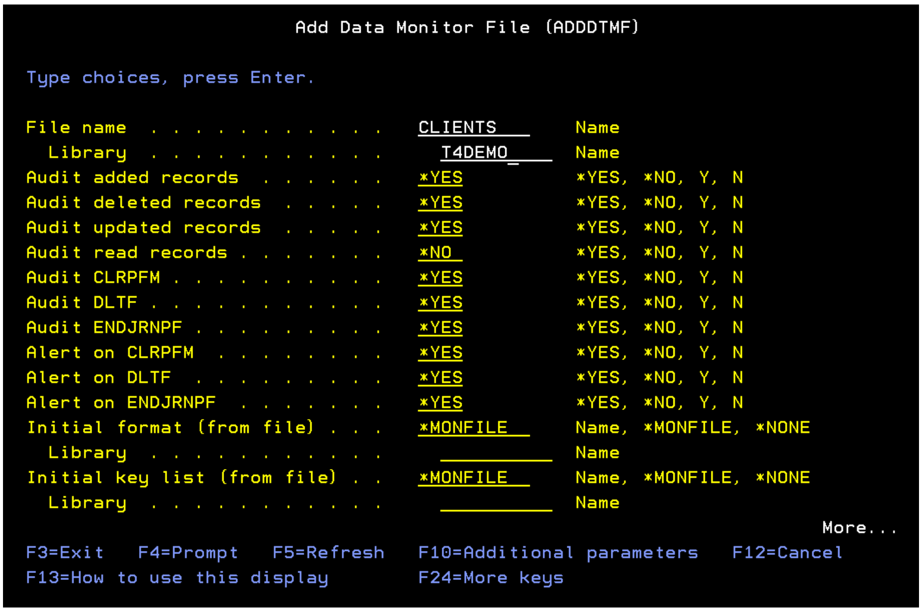
> "If you have selected to use the MINENTDTA parameter (i.e.*FILE) for the journal, then some of your journal entries entry-specific data will be minimized. The layout of the minimized record changes entries is completely different than the layout when the entry is not minimized. The data is not even recognizable nor is it readable as sophisticated hash techniques are used in addition to only operating on actual changed bytes. Therefore, if you want to use the journal as an audit mechanism, you may not want to choose this option (i.e. *FILE) for database physical files since you (i.e. Data Monitor) will not be able to read the actual change made."

See commands CRTJRN and CHGJRN for more details.

To add a new file to Data Monitor press F6 from the Work with monitored files screen. Another way to add a new file is to enter the following command:

```
T4DATAMON/ADDDTMF
```

The add Data Monitor file command (ADDDTMF) adds a file configuration entry to the configuration list. The following screen appears:



```
                        Add Data Monitor File (ADDDTMF)

 Type choices, press Enter.

 File name  . . . . . . . . . . .   CLIENTS        Name
   Library  . . . . . . . . . . .     T4DEMO       Name
 Audit added records  . . . . . .   *YES           *YES, *NO, Y, N
 Audit deleted records  . . . . .   *YES           *YES, *NO, Y, N
 Audit updated records  . . . . .   *YES           *YES, *NO, Y, N
 Audit read records . . . . . . .   *NO            *YES, *NO, Y, N
 Audit CLRPFM . . . . . . . . . .   *YES           *YES, *NO, Y, N
 Audit DLTF . . . . . . . . . . .   *YES           *YES, *NO, Y, N
 Audit ENDJRNPF . . . . . . . . .   *YES           *YES, *NO, Y, N
 Alert on CLRPFM  . . . . . . . .   *YES           *YES, *NO, Y, N
 Alert on DLTF  . . . . . . . . .   *YES           *YES, *NO, Y, N
 Alert on ENDJRNPF  . . . . . . .   *YES           *YES, *NO, Y, N
 Initial format (from file) . . .   *MONFILE       Name, *MONFILE, *NONE
   Library  . . . . . . . . . . .                  Name
 Initial key list (from file) . .   *MONFILE       Name, *MONFILE, *NONE
   Library  . . . . . . . . . . .                  Name
                                                                     More...
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
```

Figure 8 – Add Data Monitor File screen I

Enter the *File name* and *Library* of the journaled file you want to monitor.

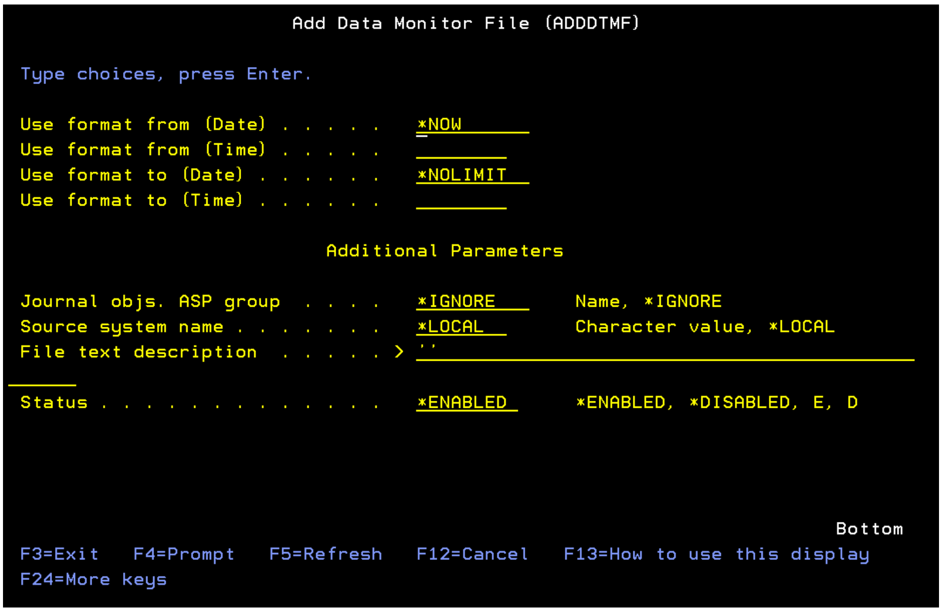
Next select the transactions you would like to monitor by setting each parameter to *YES or *NO.

> **Note**
> By default the flag Audit read records is set to *NO.

You can select to monitor both auditing values and alerts.

Indicate the *Initial format* to use and the *Library* where it can be found, in addition to the *Initial key list* and *Library*. Note that you can add more formats and keys at a later date. These concepts and how Data Monitor uses them are described in *see section 3.4 - Work With File Formats on page 17*.

Press PgDn and the second configuration screen for configuring new files appears:



```
                          Add Data Monitor File (ADDDTMF)

 Type choices, press Enter.

 Use format from (Date) . . . . .   *NOW
 Use format from (Time) . . . . .
 Use format to (Date) . . . . . .   *NOLIMIT
 Use format to (Time) . . . . . .

                             Additional Parameters

 Journal objs. ASP group  . . . .   *IGNORE       Name, *IGNORE
 Source system name . . . . . . .   *LOCAL        Character value, *LOCAL
 File text description  . . . . . > ''

 Status . . . . . . . . . . . . .   *ENABLED      *ENABLED, *DISABLED, E, D




                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 9 – ADDDTMF configuration screen 2 showing additional parameters (F10)

On the second ADDDTMF screen complete the configuration of the valid date and time for this format and keys.

Use format from Date/Time: Here you can enter the start of the date range during which the format is valid. You can enter a date or one of the valid values:

- ***NOW**: Start the format from the current system date.
- ***NOLIMIT**: Format will always be applied. If this value is indicated, Data Monitor only will use the "Date To" value to determine the corresponding format.

When entering a date, use the time value to fine-tune the start of the format.

Use format to Date/Time: Enter the end of the date range during which the format is valid. You can enter a date or one of the valid values:

- ***NOW**: End the format from the current system date.
- ***NOLIMIT**: Format will always be applied. If this value is indicated, Data Monitor only will use the "Date From" value to determine the corresponding format.

When entering a date, use the time value to fine-tune the end of the format.

You can choose to change the additional parameters or simply use the default settings. They are not mandatory for Data Monitor to work.

By default new files appear in the Work with monitored files screen with Monitor Status *INACTIVE, Enabled *YES, and the values for Latest record date in historical are blank. When you start the monitor (F10) The Monitor Status will change to *ACTIVE.

Sometimes the Latest Record Date in Historical field is filled immediately with an earlier date. This means that this file has been monitored in the past and for whatever reason the configuration entry was removed. However the monitoring data from the previous monitor was stored and now appeared when you started the monitor.

© 2013 Tango/04 Computing Group

Important

To start monitors successfully, make sure that the user profile of the Data Monitor jobs has access to the files added to Data Monitor. This user profile must also have `*ALL` authority for the journal object and `*USE` authority for journal receiver objects.

The default user profile is `QPGMR`, but you can change it to add authorization for the required files. Use the `CHGJOBD` command in job descriptions `T4DATAMON/DTMMONITOR` and `T4DATAMON/STRDTAMON`.

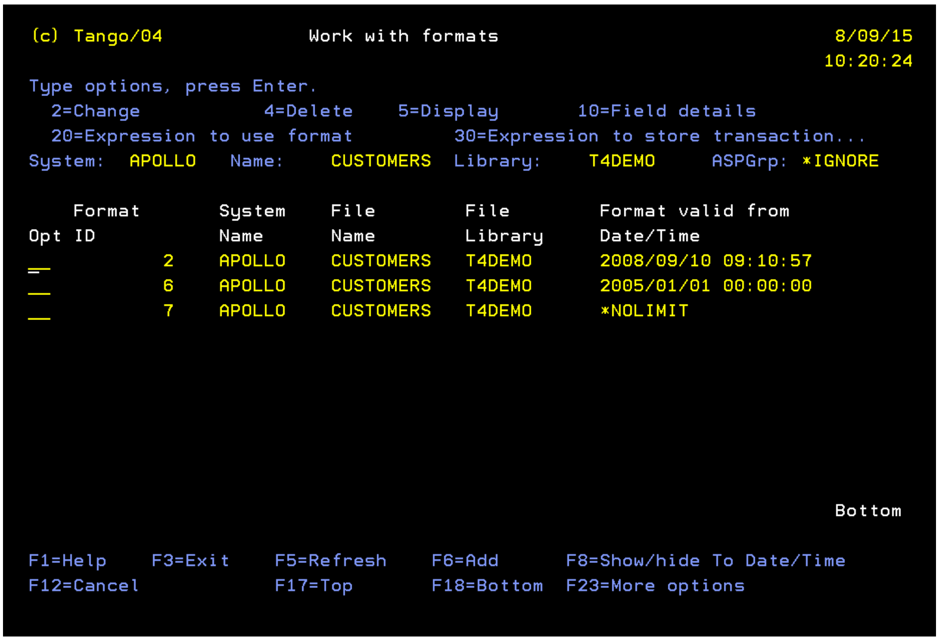
3.4 Work With File Formats

Transaction entries in IBM journals store raw (unformatted) data. Data Monitor retrieves this data and formats it for storage in the historical transaction log according to the configuration. Therefore you must specify a format for Data Monitor to use when monitoring sensitive files.

A format (or schema) includes a list of field definitions for the sensitive file and a list of key fields for ordering the records in transaction log reports. These field and key definitions translate the raw data stored in IBM Journals to records Data Monitor can read and store in its historical transaction log.

You can add different formats for the same sensitive file based on a date and time. Formats have a starting and ending date/time that defines during which period the format is valid. That way, Data Monitor can keep track of the format changes over time and process all transactions, regardless of the date they occurred.

In addition Data Monitor supports multi-format files, that is, it allows more than one valid format during the same time interval. For details on how to use multi-format files in data monitor *see section 3.4.3 - Using expressions in formats on page 21* and *see section 3.4.1 - Adding a Format to a File on page 19*. To view current format definitions of a file in Data Monitor, select option **20** from the Work with monitored files screen. The Work with formats screen appears displaying any configured file formats.



```
(c) Tango/04              Work with formats                          8/09/15
                                                                    10:20:24
Type options, press Enter.
  2=Change           4=Delete    5=Display       10=Field details
  20=Expression to use format         30=Expression to store transaction...
System:  APOLLO   Name:    CUSTOMERS  Library:    T4DEMO     ASPGrp: *IGNORE

    Format       System    File        File        Format valid from
Opt ID           Name      Name        Library     Date/Time
__          2    APOLLO    CUSTOMERS   T4DEMO      2008/09/10 09:10:57
__          6    APOLLO    CUSTOMERS   T4DEMO      2005/01/01 00:00:00
__          7    APOLLO    CUSTOMERS   T4DEMO      *NOLIMIT







                                                                    Bottom

F1=Help    F3=Exit    F5=Refresh    F6=Add     F8=Show/hide To Date/Time
F12=Cancel            F17=Top       F18=Bottom F23=More options
```

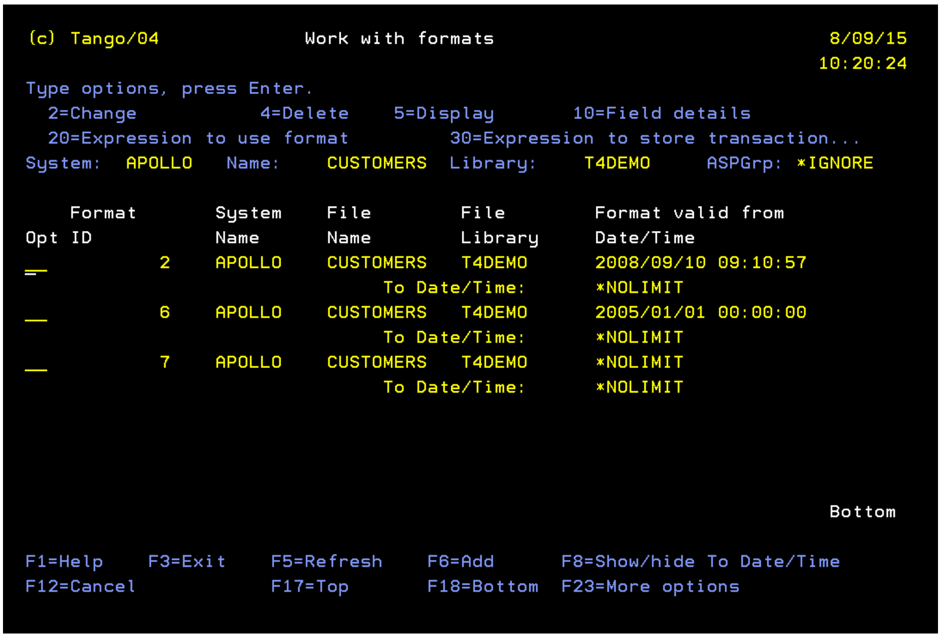
Figure 10 – Work with formats screen

Note
Not all options are visible in this screen. Use F23 to view additional options in Work with monitored files.

Beneath the options on the Work with formats screen there is a header showing the system, name, library, and ASP group of the monitored file. Brief description of each field in the header:

- **System**: Indicates the system where the file to be monitored by Data Monitor is located, in this case MUNDAKA.
- **Name**: Name of the file to be monitored.
- **Library**: Library name of the file to be monitored.
- **ASPGrp**: This value is used when searching journal and journal receiver objects where the file is journaled. The default value *IGNORE is used when you do not work with ASP groups in your system.

Use F8 for quick insight into a formats date and time definitions, as shown in the following figure.



```
 (c)  Tango/04              Work with formats                          8/09/15
                                                                      10:20:24
 Type options, press Enter.
   2=Change            4=Delete     5=Display       10=Field details
   20=Expression to use format          30=Expression to store transaction...
 System:   APOLLO    Name:     CUSTOMERS  Library:     T4DEMO      ASPGrp: *IGNORE

     Format        System     File         File         Format valid from
 Opt ID            Name       Name         Library      Date/Time
 __          2     APOLLO     CUSTOMERS    T4DEMO       2008/09/10 09:10:57
                                   To Date/Time:        *NOLIMIT
 __          6     APOLLO     CUSTOMERS    T4DEMO       2005/01/01 00:00:00
                                   To Date/Time:        *NOLIMIT
 __          7     APOLLO     CUSTOMERS    T4DEMO       *NOLIMIT
                                   To Date/Time:        *NOLIMIT




                                                                       Bottom

 F1=Help     F3=Exit     F5=Refresh     F6=Add      F8=Show/hide To Date/Time
 F12=Cancel              F17=Top        F18=Bottom  F23=More options
```

Figure 11 – F8 Display date/time definition

Each transaction has a date and time. For selecting the format to apply Data Monitor takes the format with the nearest *Format valid from Date/Time* value. For a detailed explanation *see Appendix E: Formats with Overlapping Date Ranges on page 104*.

To change the default criterion of the nearest date, you can use an expression in *expression to use format* using option **20**. This expression will decide which format will be applied. If the result of this expression is true in both cases, then Data Monitor reverts to the default criterion explained above.

Note
Data Monitor uses ALEV to define the logical expressions,
For details *see Appendix G: ALEV Variables and Functions for Expressions on page 109.*

Using the example in the previous figure, a transaction done on the 24th of April 2005 will use format 6, according to the default.

However, you can change the behavior by using expressions. For example if expression for format `30` is false, format `29` will be applied.

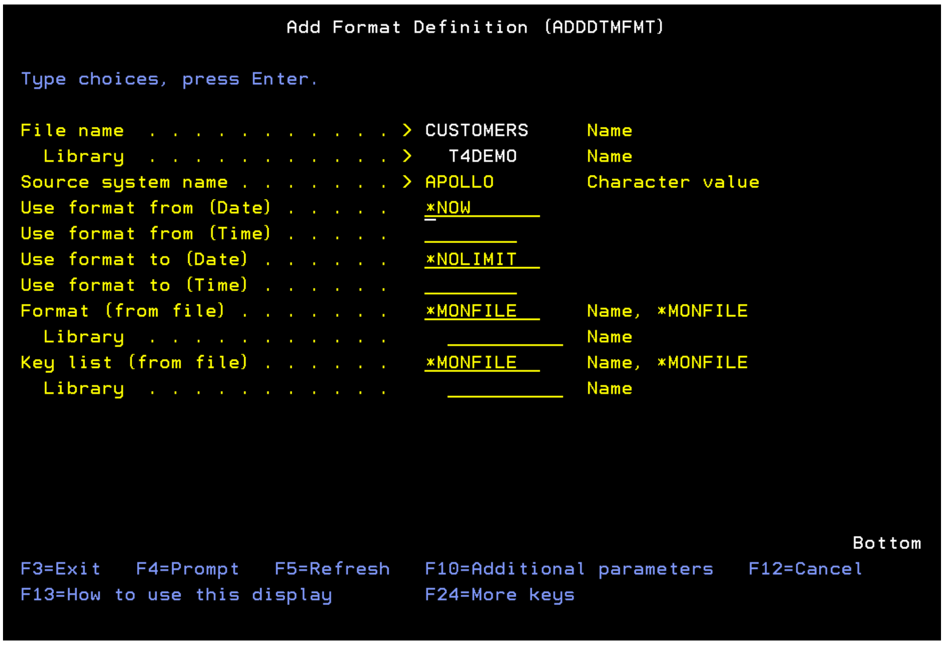
3.4.1 Adding a Format to a File

To add a format definition to a file in Data Monitor, select option **20** from the Work with monitored files screen. The Work with formats screen appears displaying any configured file formats.

To add a new format, select F6 from the Work with formats screen. Another way to add a format is to enter the command:

```
T4DATAMON/ADDDTMFMT
```

The Add Format Definition (`ADDDTMFMT`) screen appears:



```
                     Add Format Definition (ADDDTMFMT)

Type choices, press Enter.

File name  . . . . . . . . . . . > CUSTOMERS     Name
  Library  . . . . . . . . . . . >   T4DEMO      Name
Source system name . . . . . . . > APOLLO        Character value
Use format from (Date) . . . . .   *NOW
Use format from (Time) . . . . .
Use format to (Date) . . . . . .   *NOLIMIT
Use format to (Time) . . . . . .
Format (from file) . . . . . . .   *MONFILE      Name, *MONFILE
  Library  . . . . . . . . . . .                 Name
Key list (from file) . . . . . .   *MONFILE      Name, *MONFILE
  Library  . . . . . . . . . . .                 Name





                                                                     Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 12 – Add Format Definition (ADDDTMFMT) screen

Enter the **File name** and **Library** of the sensitive file to which the format applies.

Source system name: name of the system where the file originated.

Enter the **starting and ending dates and times for the file format.** If you want the format to apply always, leave the default settings as displayed in *Figure 12*.

Enter the File name and Library of the *format* to apply. Special value `*MONFILE`, means that Data Monitor will extract the format from the file you are monitoring (parameters *File name* and *Library*). Note that this file must exist in the system when adding the format.

Enter the File name and the Library of the *key list* to apply. Special value `*MONFILE`, means that Data Monitor will extract the key fields list from the file you are monitoring (parameters *File name* and *Library*). For example, you can enter a logical file you use to access the file being monitored and you will see transaction in reports in the sequence provided by this logical file. This file must exist in the system when adding the key field list.

Note
In case you create an auxiliary file for specifying the key list, once a format has been added to the configuration (see above) Data Monitor no longer needs to keep the file in the system.

Press F10 for *additional parameters*. Two more fields appear on the screen and you can now access additional screens with parameters pressing PgDn.

In additional parameters you will find the following fields:

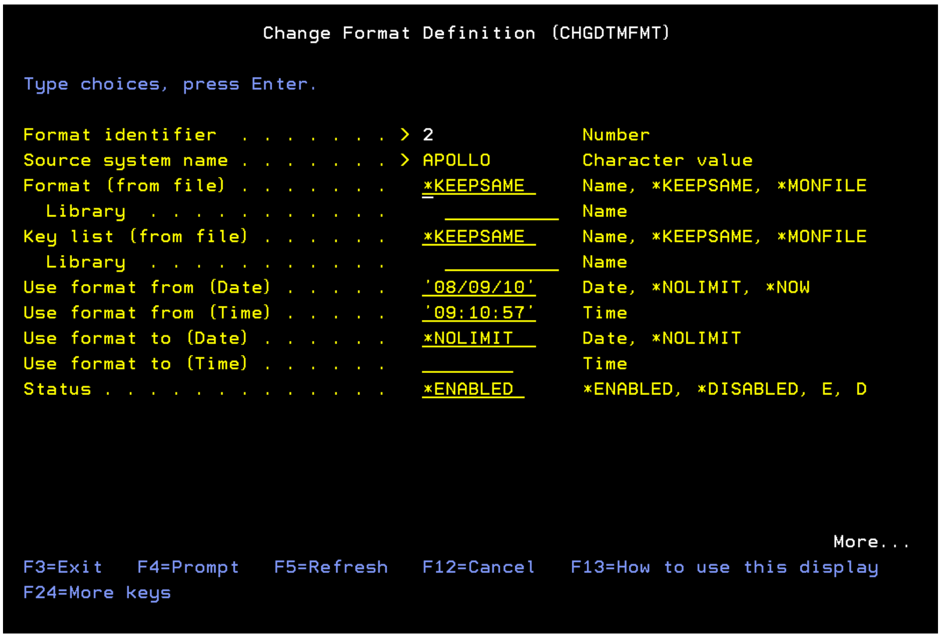
- **Journal Objects ASP Group**: (only for V5R2M0 and later) This value is used when searching journal and journal receiver objects where file is journaled. If you were not aware of ASP groups, or do not work with ASP groups in your system, ignore this parameter and set it to the default special value `*IGNORE`. If you work with ASP groups enter the ASP group name here.
- **Status**: the user can enable or disable the entry. The only valid values in this field are `*ENABLED` or `*DISABLED`. When the user executes function F10 (start and reinitiate all monitors) only the enabled entries will be started or reinitialized.
- **Format text description**: Text description of the format file. The user can enter a specific value or select `*FMTFILE`, in which case Data Monitor will retrieve the current text description from the format file. The format file description has a maximum length of 50 characters.

3.4.2 Changing Format Definitions

To modify existing format definitions select option **20** from the Work with monitored files screen. The Work with formats screen appears displaying the configured file formats. Select option **2** to change a file format. Another way to change a format is to enter the command:

`T4DATAMON/CHGDTMFMT`. Press F4 and select **Format ID**.

The Change Format Definition (`CHGDTMFMT`) screen appears:



```
                    Change Format Definition (CHGDTMFMT)

 Type choices, press Enter.

 Format identifier . . . . . . . > 2             Number
 Source system name . . . . . . . > APOLLO        Character value
 Format (from file) . . . . . . .   *KEEPSAME      Name, *KEEPSAME, *MONFILE
   Library . . . . . . . . . . .                  Name
 Key list (from file) . . . . . .   *KEEPSAME      Name, *KEEPSAME, *MONFILE
   Library . . . . . . . . . . .                  Name
 Use format from (Date) . . . . .   '08/09/10'     Date, *NOLIMIT, *NOW
 Use format from (Time) . . . . .   '09:10:57'     Time
 Use format to (Date) . . . . . .   *NOLIMIT       Date, *NOLIMIT
 Use format to (Time) . . . . . .                  Time
 Status . . . . . . . . . . . . .   *ENABLED       *ENABLED, *DISABLED, E, D




                                                                    More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 13 – Change Format Definition (CHGDTMFMT) screen

Press page down for more parameters.

The **Format ID** is an internal identifier for Data Monitor.

Source system name is the name of the system where the file originated. You can enter a specific name or the default value `*LOCAL` to use the local system.

Format: file name for the file in which we are making a format change. Data Monitor offers dynamic and automatic management of formats for the files it is monitoring. The user can enter either a specific value or one of the following special values:

- `*KEEPSAME`: With this value, no change will be made to what is already stored.
- `*MONFILE`: This value indicates that the name of the format file is the same as the file being monitored.

If the user enters a specific value the current format will be recreated. This parameter is mandatory.

In the field **Key list** enter the name of the key field format file for the previously selected file. Data Monitor offers dynamic and automatic management of formats for the files it is monitoring. The user can enter either a specific value or one of the following special values:

- `*KEEPSAME`. With this value, no change will be made to what is already stored.
- `*MONFILE`. This value indicates that the name of the key field format file is the same as the file being monitored.

This parameter is mandatory. If the user enters a specific value the current key field formats will be recreated.

Note

Any changes to the format file configuration entry will take effect immediately in monitors that read journals and save data in the historical transaction log. Transactions processed previously to the change will keep the format that applied when they were processed. To apply the new format to transactions that were processed with an earlier format *see Appendix C: How to Re-Process Entries on page 99.*

Status: The user can enable or disable the current entry. The only possible values are:

- `*ENABLED`. With this value we enable the current entry.
- `*DISABLE`. With this value we disable the current configuration entry.

3.4.3 Using expressions in formats

There are three types of expressions you can use in formats. All three options are available from the work with formats menu:

- Option **20** - Expression to use format (`USEIF`)
- Option **30** - Expression to store transaction (`STOREIF`)
- Option **40** - Preprocess expression (`PREPROC`)

Expression to use format (option 20)

Sometimes a user may need to use more than one valid format for the same time period – i.e. what we call *multi-format files*. For example, you might need to use a different record format for the same table, depending on the value of a certain field. The expression `USEIF` allows you to enter the checks and comparisons required to determine what format to use. The expression should be a logical expression and if this logical expression returns a True value, this format will be used.

Note that every transaction for a file being monitored must have at least one valid format. If not, a warning message will be sent to the message queue for every transaction (insert, delete, update, read).

Expression to store (option 30)

If you want to store only a selection of transactions (for example, those where the client type is *important*) use this field to introduce conditional expressions that must be met for storing a transaction. The expression should be a logical expression and if this logical expression returns a True value, the transaction will be stored.

The conditional expression `STOREIF` is only evaluated after the definition of this transaction format has been applied (including checking the validity period of the format and the checks of the conditional expression `USEIF`).

Note
The expression to store applies to the audit event and not to the alert. Thus it is possible to send Alerts for audit events, even though the audit event is not stored in the database.

Preprocess expression (option 40)

To display additional options press F23 – More options. Option **40** – Preprocess expression will appear.

The user can enter an expression for pre-processing data and variables of a transaction before storing it. If the field is left blank no changes are made to the original data of the transaction.

The preprocess is an excellent and powerful way to enrich data, to get data from existing fields (such as creating virtual fields by adding two fields, or concatenating various fields of character chain or string type), or to get additional information that is not found in the original records (for example application-specific data).

In data update transactions the fields are available in their original state (vector BeforeValues) and in their changed state (vector AfterValues).

For example, to get the sum of the numerical changes to fields A and B and use it to change C, enter:

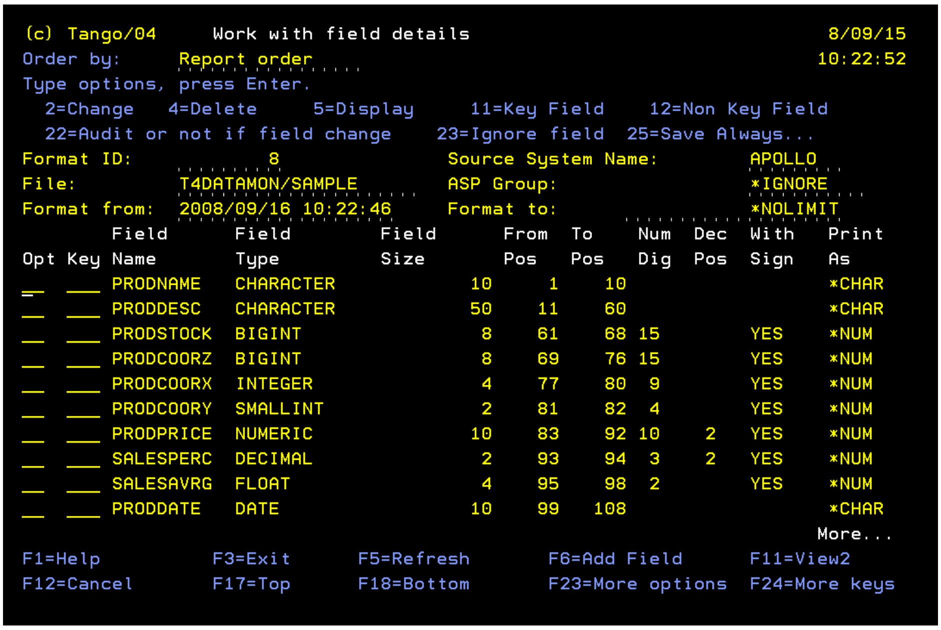
```
&AfterValues[“C”]:=&AfterValues[“A”] + AfterValues[“B”]
```

Note
Data Monitor uses ALEV to define the logical expressions *see Appendix G: ALEV Variables and Functions for Expressions on page 109*. For a more detailed description of how expressions are used in processing journal entries for a monitored file *see Appendix B: Journal Entry Process Flow on page 95.*

Use option **10** – Field details to work with file format details.

3.5 Work With File Format Details

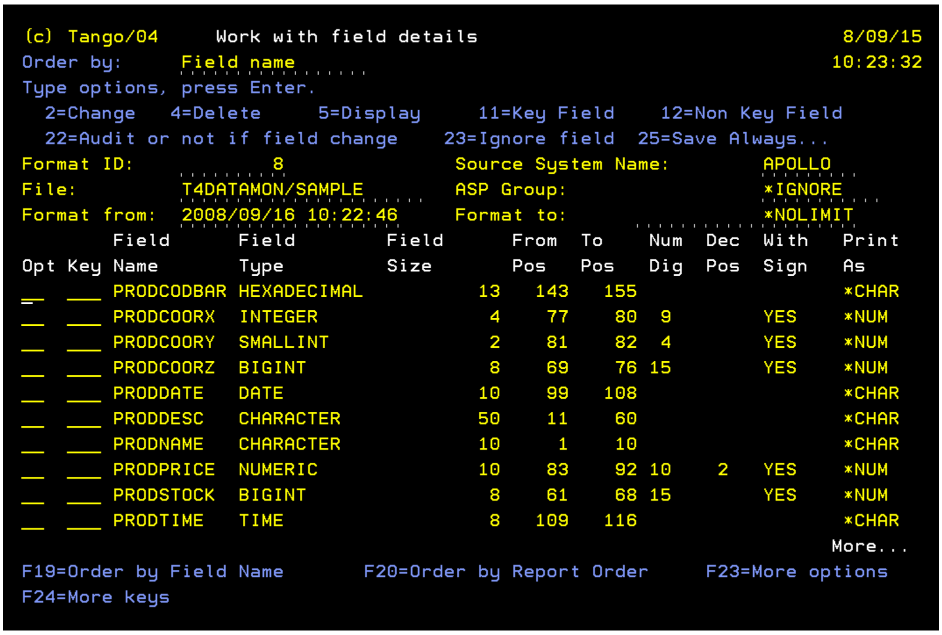
For an in-depth view of the format details use option **10** – Field details and the Work with field details screen appears.



```
(c) Tango/04     Work with field details                            8/09/15
Order by:      Report order                                        10:22:52
Type options, press Enter.
  2=Change    4=Delete     5=Display     11=Key Field     12=Non Key Field
  22=Audit or not if field change    23=Ignore field  25=Save Always...
Format ID:             8               Source System Name:         APOLLO
File:          T4DATAMON/SAMPLE        ASP Group:                  *IGNORE
Format from:   2008/09/16 10:22:46     Format to:                  *NOLIMIT
        Field      Field        Field       From  To    Num  Dec  With   Print
Opt Key Name       Type         Size         Pos  Pos   Dig  Pos  Sign   As
__  ___ PRODNAME   CHARACTER            10     1    10                   *CHAR
__  ___ PRODDESC   CHARACTER            50    11    60                   *CHAR
__  ___ PRODSTOCK  BIGINT                8    61    68 15         YES    *NUM
__  ___ PRODCOORZ  BIGINT                8    69    76 15         YES    *NUM
__  ___ PRODCOORX  INTEGER               4    77    80  9         YES    *NUM
__  ___ PRODCOORY  SMALLINT              2    81    82  4         YES    *NUM
__  ___ PRODPRICE  NUMERIC              10    83    92 10     2   YES    *NUM
__  ___ SALESPERC  DECIMAL               2    93    94  3     2   YES    *NUM
__  ___ SALESAVRG  FLOAT                 4    95    98  2         YES    *NUM
__  ___ PRODDATE   DATE                 10    99   108                   *CHAR
                                                                        More...
F1=Help          F3=Exit      F5=Refresh       F6=Add Field      F11=View2
F12=Cancel       F17=Top      F18=Bottom       F23=More options  F24=More keys
```

Figure 14 – File format details by report order

By default the fields are sorted by report order. Report order is the order in which the fields appear in the Data Monitor Report in Reports. Use F24 – More keys to display additional functions. Use F19 – Order by field name or F20 – Order by report order if you want to change the sort order on screen.



```
(c) Tango/04     Work with field details                             8/09/15
Order by:      Field name                                           10:23:32
Type options, press Enter.
  2=Change    4=Delete     5=Display     11=Key Field     12=Non Key Field
  22=Audit or not if field change    23=Ignore field  25=Save Always...
Format ID:             8               Source System Name:          APOLLO
File:          T4DATAMON/SAMPLE        ASP Group:                   *IGNORE
Format from:   2008/09/16 10:22:46     Format to:                   *NOLIMIT
        Field       Field        Field       From  To    Num  Dec  With   Print
Opt Key Name        Type         Size         Pos  Pos   Dig  Pos  Sign   As
__  ___ PRODCODBAR  HEXADECIMAL          13   143   155                   *CHAR
__  ___ PRODCOORX   INTEGER               4    77    80  9         YES    *NUM
__  ___ PRODCOORY   SMALLINT              2    81    82  4         YES    *NUM
__  ___ PRODCOORZ   BIGINT                8    69    76 15         YES    *NUM
__  ___ PRODDATE    DATE                 10    99   108                   *CHAR
__  ___ PRODDESC    CHARACTER            50    11    60                   *CHAR
__  ___ PRODNAME    CHARACTER            10     1    10                   *CHAR
__  ___ PRODPRICE   NUMERIC              10    83    92 10     2   YES    *NUM
__  ___ PRODSTOCK   BIGINT                8    61    68 15         YES    *NUM
__  ___ PRODTIME    TIME                  8   109   116                   *CHAR
                                                                         More...
F19=Order by Field Name       F20=Order by Report Order      F23=More options
F24=More keys
```

Figure 15 – File format details by field name

Below the options you will find a header containing the Format ID, File, Source System Name, ASP group, and Format From and To definition. Here follows a brief description of each field:

Format ID: This is the format identifier of each format file related to a file monitored by Data Monitor.

File: Format file name for monitored files.

Source system name: The user can select the source system where the file format related to the file to be monitored by Data Monitor is located.

ASP group: This value is used when searching journal and journal receiver objects where file is journaled. The following special value is used if you do not work with ASP groups in your system:

```
*IGNORE
```

Format From/To: Each format file configuration entry has two related values: *Date From* and *Date To* values. For each format there is a date and time indicating the beginning and the end of this range.

Data Monitor will use these values and the current system date and time values to determine what format to apply. If the current system date and time values are included in the range, then Data Monitor will apply the corresponding format.

You can use option **2** – Change in any of the following views to change the format details.

Note

Any changes to the format file configuration entry will take effect immediately in monitors that read journals and save data in the transaction log. Transactions processed previously to the change will keep the format that applied when they were processed. To apply the new format to transactions that were processed with an earlier format *see Appendix C: How to Re-Process Entries on page 99.*

3.5.1 View 1- Basic Field Definitions

There are three views of Work with field details. *Figure 14* shown above is the default view.

View 1 shows the basic field definitions.

Field name: This is the field name for a specified format file configured in Data Monitor. There are no duplicate fields as the fields displayed here are related to a specific format identifier and a specific system.

Field type: This is the field type. Possible values are:

Value	Description
CHARACTER	The field contains EBCDIC character data.
DECIMAL	The field contains packed decimal numbers with an implicit decimal point.
NUMERIC	The field contains zoned decimal numbers with an implicit decimal point.
BIGINT	The field contains a binary number with a precision of 63 bits.
INTEGER	The field contains a binary number with a precision of 31 bits.
SMALLINT	The field contains a binary number with a precision of 15 bits.
FLOAT	The field contains floating point numbers in IEEE format.
HEXADECIMAL	The field contains a string in hexadecimal format.
DATE	The field contains a value in date format.
TIME	The field contains a value in time format.
TIMESTAMP	The field contains a seven-part value that designates a date and time under the Gregorian calendar. The seven parts are year, month, day, hour, minute, second, and microsecond.

Value	Description
CHARACTER-O	The field can contain both DBCS and SBCS character data.
CHARACTER-E	The field contains either DBCS or SBCS character data.
CHARACTER-J	The field contains only DBCS character data.
GRAPHIC	The field contains fixed-length graphic data.
BINARYCHAR	The field contains a string in binary format.

Field size: This value represents the length of the selected field.

From Pos: This is the initial position for the field in file format you are configuring for Data Monitor.

To Pos: This is the final position for the field in the file format you can configure for Data Monitor.

Num Dig: This is the number of digits for this field. This value is only relevant for numeric field types.

Dec Pos: This is the number of decimal positions in the field. This value is only relevant for numeric field types.

With Sign: Use this value to indicate whether the field is a signed field or not. If this value is YES then the field is a signed field.

Print As: Representation mode for this field. Possible values are:

- *NUM: This field is interpreted like numerical field.
- *CHAR: This field is interpreted like character field.

Note that not all options are displayed on this screen. Press F23 for additional options.

Use option **2** to change any of the details shown in this view.

Key Fields

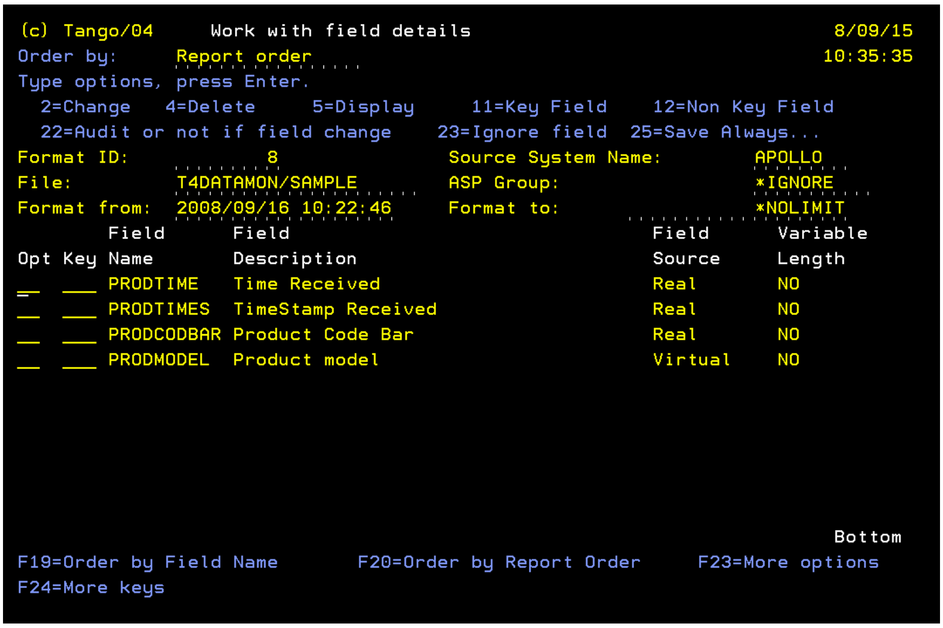
It is possible to indicate key fields in Data Monitor. To change a *field* to a *key field* in a file use option **11** – Key field and Data Monitor will automatically add the key. You can also change the key manually: simply enter the key number in the key field of the required field. To remove a key, first position on the field where you want to remove the key. Next enter option **12** – Non Key Field.

Once you add a key field, the key fields will be displayed at the top of the list of fields. The remaining non-key fields will then be sorted according to the sort order you indicated (i.e. report order or field name).

3.5.2 View 2 - Field Description, Source and Variable Length

Use F11 - View 2 to open the second view.

The second view displays the full description of the field, real or virtual field source and indicates whether the field length is variable.



```
(c) Tango/04      Work with field details                            8/09/15
Order by:      Report order . . . .                                 10:35:35
Type options, press Enter.
  2=Change    4=Delete     5=Display     11=Key Field     12=Non Key Field
  22=Audit or not if field change    23=Ignore field  25=Save Always...
Format ID:     . . . . . . . 8              Source System Name:        APOLLO . .
File:          T4DATAMON/SAMPLE . . . .     ASP Group:                 *IGNORE . . .
Format from:   2008/09/16 10:22:46          Format to:        . . . . . *NOLIMIT
        Field      Field                                   Field      Variable
Opt Key Name       Description                             Source     Length
__  ___ PRODTIME   Time Received                           Real       NO
__  ___ PRODTIMES  TimeStamp Received                      Real       NO
__  ___ PRODCODBAR Product Code Bar                        Real       NO
__  ___ PRODMODEL  Product model                           Virtual    NO







                                                                     Bottom
F19=Order by Field Name      F20=Order by Report Order     F23=More options
F24=More keys
```

Figure 16 – View 2 of work with field details

Field description

This value is the TEXT value in the field definition when you have defined it when you have created (for example, using DDS) or if TEXT value is an empty string, then this value is the COLHDG (column headings) value.

Field Source

A Real field is one that was created automatically when the format was created. A Virtual field is introduced manually by the user. The initial and final positions of a virtual field may overlap real and other virtual fields.

For example, our Clients database has the following real fields:

Field	Position
Street	1-40
City	40-55
Province	55-70
Postal code	70-75

We can create a virtual field to show all details in one field by defining the following field positions: 1..75

In reports no distinction is made between real and virtual fields.

⚠ **Important**
We strongly recommend you do NOT change a virtual field to a real field, even though it is possible to change it using option **2**=Change for this field. This prevents problems with the automatic refresh function for formats of changed files (for example where fields were added, deleted, or field attributes were changed).

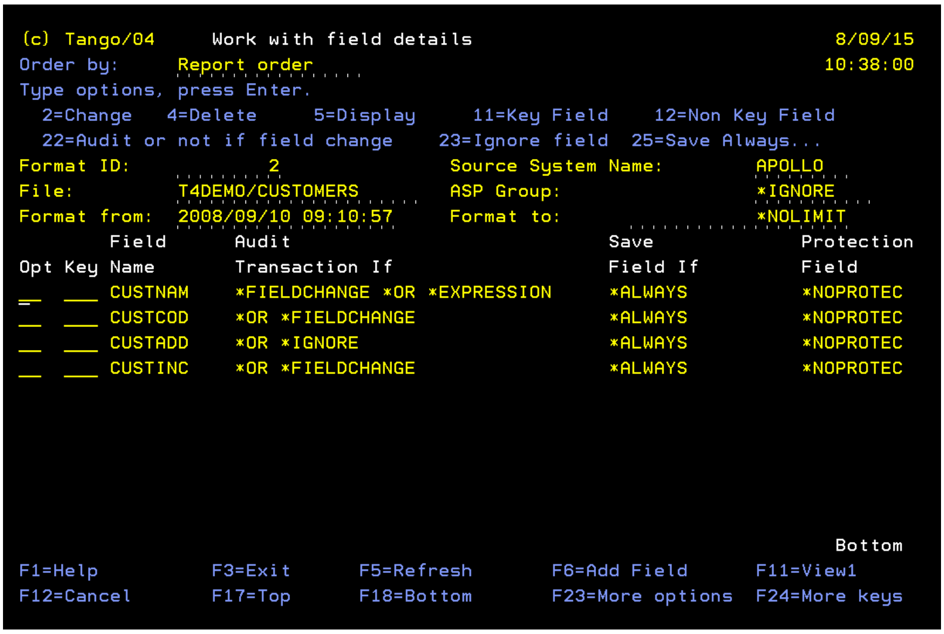
For more details *see Chapter 6 - Handling formats in Data Monitor for iSeries on page 58.*

Variable length

This value indicates if the selected field is a variable length field or not.

3.5.3 View 3 - Auditing and Protection Details

Use F11 – View 3 to open the third view. Here you will find the auditing and protection details for each field.



```
(c) Tango/04     Work with field details                                 8/09/15
Order by:     Report order                                               10:38:00
Type options, press Enter.
  2=Change   4=Delete     5=Display     11=Key Field     12=Non Key Field
  22=Audit or not if field change    23=Ignore field  25=Save Always...
Format ID:            2                  Source System Name:        APOLLO
File:         T4DEMO/CUSTOMERS           ASP Group:                 *IGNORE
Format from:  2008/09/10 09:10:57        Format to:                 *NOLIMIT
        Field      Audit                            Save             Protection
Opt Key Name       Transaction If                   Field If         Field
__  ___ CUSTNAM    *FIELDCHANGE *OR *EXPRESSION     *ALWAYS          *NOPROTEC
__  ___ CUSTCOD    *OR *FIELDCHANGE                 *ALWAYS          *NOPROTEC
__  ___ CUSTADD    *OR *IGNORE                      *ALWAYS          *NOPROTEC
__  ___ CUSTINC    *OR *FIELDCHANGE                 *ALWAYS          *NOPROTEC





                                                                        Bottom
F1=Help          F3=Exit      F5=Refresh       F6=Add Field      F11=View1
F12=Cancel       F17=Top      F18=Bottom       F23=More options  F24=More keys
```

Figure 17 – View 3 Work with field details

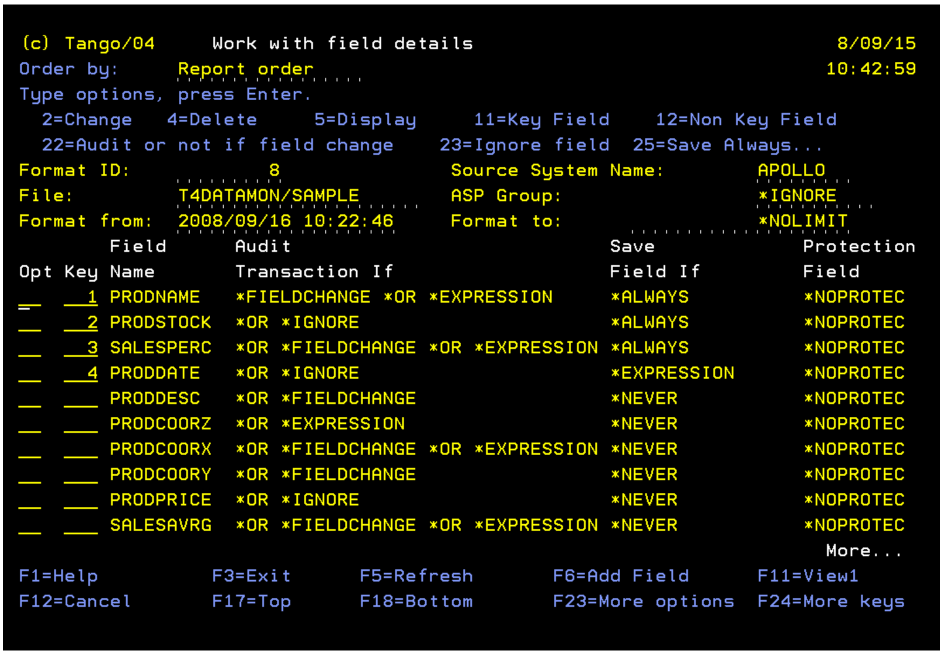
Remember that Data Monitor can use expressions in formats (expression to store) to decide if a transaction must be audited or not, *see section 3.4.3 - Using expressions in formats on page 21*. If the expression to store, as defined in work with formats, is true, the conditions at field level must also be evaluated.

At field level Audit Transaction If is used to determine whether to save the transaction. Audit Transaction If has three possible values: `*FIELDCHANGE`, `*EXPRESSION` and `*IGNORE`. But these values are for all the fields in this format.

Together they build a unique expression to evaluate if it is true or not. With this result Data Monitor can finally decide whether to audit the transaction. For example look at the figure above. Here the transaction will be audited only if:

- The value of the field `CUSTNAM` changes, or
- The expression for field `CUSTNAM` is true, or
- The value of the field `CUSTCOD` changes, or
- The value of Field `CUSTINC` changes.

It is processed as a single condition, as if you were reading column *Audit Transaction If* from top to bottom.



```
 (c) Tango/04     Work with field details                              8/09/15
Order by:     Report order                                            10:42:59
Type options, press Enter.
  2=Change   4=Delete    5=Display     11=Key Field    12=Non Key Field
  22=Audit or not if field change    23=Ignore field  25=Save Always...
Format ID:            8               Source System Name:        APOLLO
File:         T4DATAMON/SAMPLE        ASP Group:                 *IGNORE
Format from:  2008/09/16 10:22:46     Format to:                 *NOLIMIT
        Field      Audit                            Save             Protection
Opt Key Name       Transaction If                   Field If         Field
_    _1 PRODNAME   *FIELDCHANGE *OR *EXPRESSION     *ALWAYS          *NOPROTEC
__   _2 PRODSTOCK  *OR *IGNORE                      *ALWAYS          *NOPROTEC
__   _3 SALESPERC  *OR *FIELDCHANGE *OR *EXPRESSION *ALWAYS          *NOPROTEC
__   _4 PRODDATE   *OR *IGNORE                      *EXPRESSION      *NOPROTEC
__  ___ PRODDESC   *OR *FIELDCHANGE                 *NEVER           *NOPROTEC
__  ___ PRODCOORZ  *OR *EXPRESSION                  *NEVER           *NOPROTEC
__  ___ PRODCOORX  *OR *FIELDCHANGE *OR *EXPRESSION *NEVER           *NOPROTEC
__  ___ PRODCOORY  *OR *FIELDCHANGE                 *NEVER           *NOPROTEC
__  ___ PRODPRICE  *OR *IGNORE                      *NEVER           *NOPROTEC
__  ___ SALESAVRG  *OR *FIELDCHANGE *OR *EXPRESSION *NEVER           *NOPROTEC
                                                                       More...
F1=Help          F3=Exit      F5=Refresh       F6=Add Field      F11=View1
F12=Cancel       F17=Top      F18=Bottom       F23=More options  F24=More keys
```

Figure 18 – Work with field details

Audit transaction if. The default value is *FIELDCHANGE. Use option **22** to indicate whether you want to audit the transaction if there is a change to the field.

To ignore changes to the field use option **23** Ignore field and *IGNORE will appear.

Use option **30** (enter audit transaction if) to configure the condition based on which a transaction should be audited. *EXPRESSION will appear in the audit transaction if field.

It is possible for this field to contain both *FIELDCHANGE and *EXPRESSION.

Save field if. Use option **25** to always save the field (*ALLWAYS appears), or option **41** to never store the field (*Never is displayed). Use option **40** to enter an expression based on which the field should be stored (Save field if shows *EXPRESSION).

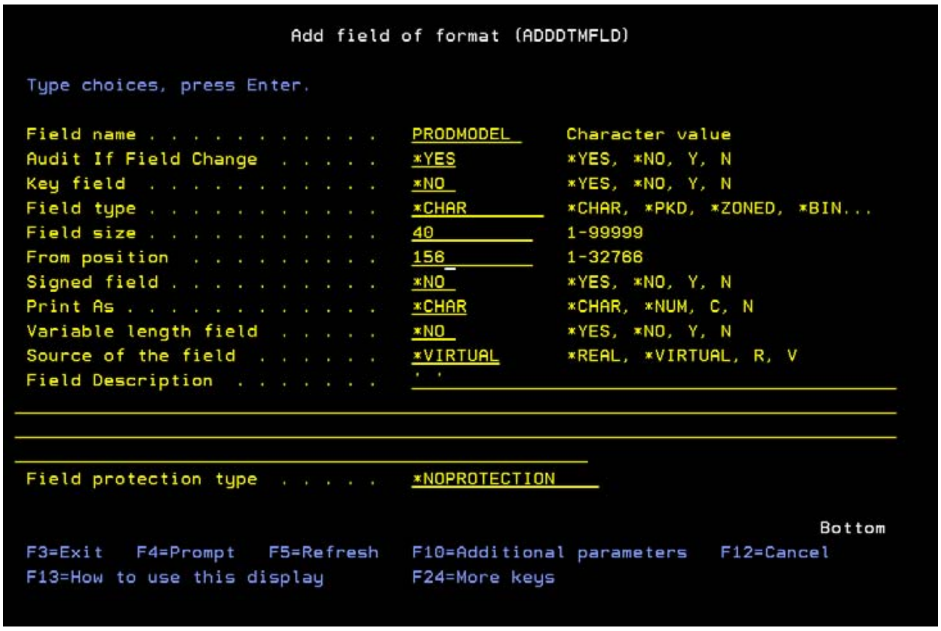
Protection field: Use option **2** change to edit the protection settings. Protection is used to protect certain confidential data. There are three possibilities:

- *NOPROTEC: the value is stored and displayed in Data Monitor reports
- *SOFT: the value is stored, but not displayed in reports. Instead of the value the report will show *Restricted*
- *HARD: the value is neither stored nor displayed in the report. The word *PROTECTED* appears in the report instead of the value.

For further details regarding protecting high level sensitive data and compliance with European Data Protection Directives (proposed at the time of publication), please *see Appendix A: Recommendations for Protecting High Level Data on page 93*.

3.5.4 Adding field of format

When you add a file from Work with monitored files you will see the default configuration of the format. To add other fields you only need to press F6 – Add field. The following screen appears.



```
                      Add field of format (ADDDTMFLD)

 Type choices, press Enter.

 Field name . . . . . . . . . . .   PRODMODEL      Character value
 Audit If Field Change  . . . . .   *YES           *YES, *NO, Y, N
 Key field  . . . . . . . . . . .   *NO            *YES, *NO, Y, N
 Field type . . . . . . . . . . .   *CHAR          *CHAR, *PKD, *ZONED, *BIN...
 Field size . . . . . . . . . . .   40             1-99999
 From position  . . . . . . . . .   156            1-32766
 Signed field . . . . . . . . . .   *NO            *YES, *NO, Y, N
 Print As . . . . . . . . . . . .   *CHAR          *CHAR, *NUM, C, N
 Variable length field  . . . . .   *NO            *YES, *NO, Y, N
 Source of the field  . . . . . .   *VIRTUAL       *REAL, *VIRTUAL, R, V
 Field Description  . . . . . . .   ' '



 Field protection type  . . . . .   *NOPROTECTION

                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
```

Figure 19 – Add a Format

Field name: This is the field name for a specified format file configured in Data Monitor.

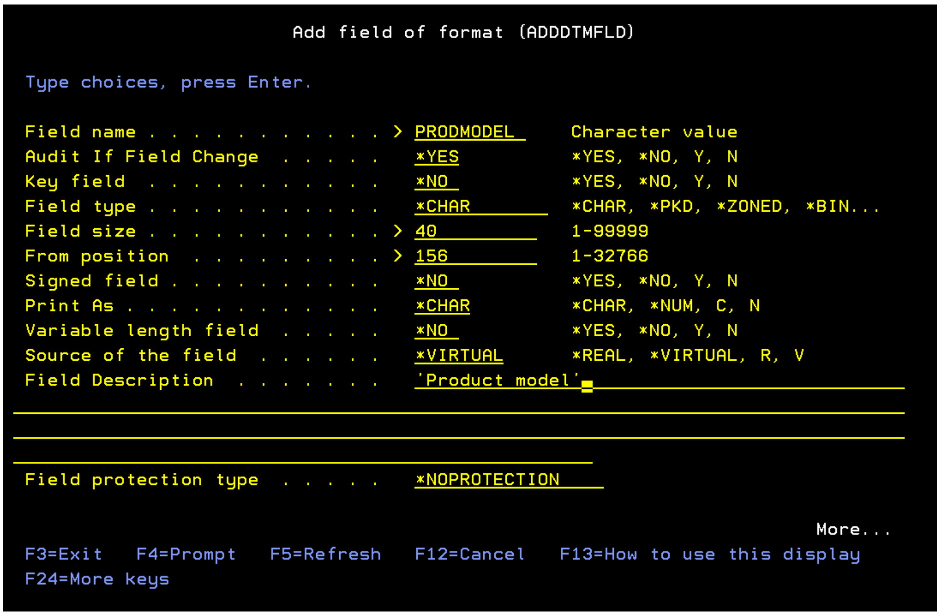
Audit if field change: This value is an arithmetic-logic expression. When Data Monitor evaluates this expression a Boolean value is returned. If this Boolean value is true then Data Monitor will audit transaction otherwise Data Monitor will not audit it.

Key field: This value indicates if the selected field is a key field or not. A positive number indicates the selected field is a key field. When the field is not a key field, then this value appears blank.

You can change this value manually by entering a positive number, which implies you are changing the selected field to a key field. And if the field you are modifying is originally a key field, you can change the order for this field when you make a report using tools included with the product. But, if you want to change a specified *key field* to a *non-key field*, then you must use option **12** - Non Key Field.

Note

If you need to change the description for the last field added you can use the option **2** – Change. The following screen appears:



```
                       Add field of format (ADDDTMFLD)

 Type choices, press Enter.

 Field name . . . . . . . . . . . > PRODMODEL      Character value
 Audit If Field Change  . . . . .   *YES           *YES, *NO, Y, N
 Key field  . . . . . . . . . . .   *NO            *YES, *NO, Y, N
 Field type . . . . . . . . . . .   *CHAR          *CHAR, *PKD, *ZONED, *BIN...
 Field size . . . . . . . . . . . > 40             1-99999
 From position  . . . . . . . . . > 156            1-32766
 Signed field . . . . . . . . . .   *NO            *YES, *NO, Y, N
 Print As . . . . . . . . . . . .   *CHAR          *CHAR, *NUM, C, N
 Variable length field  . . . . .   *NO            *YES, *NO, Y, N
 Source of the field  . . . . . .   *VIRTUAL       *REAL, *VIRTUAL, R, V
 Field Description  . . . . . . .   'Product model'


 Field protection type  . . . . .   *NOPROTECTION

                                                                     More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 20 – Change format

3.6 Starting/Restarting Monitors

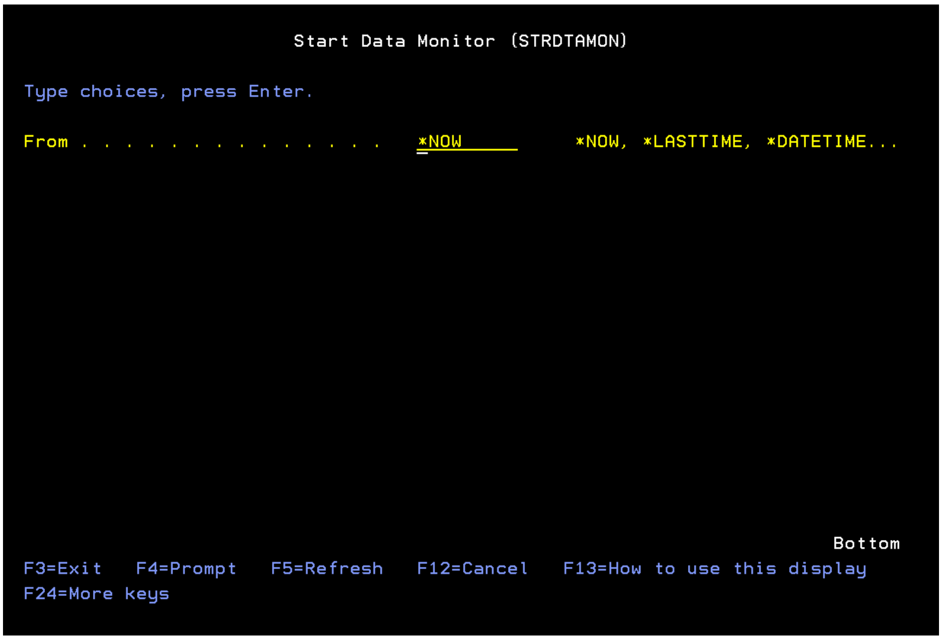
You can use F10 – Start/Restart all monitors from the Work with monitored files screen to start or restart all monitors. Another way to start/restart all monitors is to enter the following command:

```
T4DATAMON/STRDTAMON.
```

Note that only configuration entries that are enabled will be started/reinitialized when executing F10 or the command T4DATAMON/STRDTAMON.

Note also that any changes you make from the Work with monitored files screen will not take effect until monitors are restarted. Furthermore, any changes you make to ENDJRNPF or DLTF for any enabled files will not take effect until you press F10.

When using F10 – Start/Restart all monitors a screen appears where you can indicate from what time you want to start the monitor. The options are *NOW, *LASTTIME, or from *DATETIME. The monitor status of the files will change, and an *ACTIVE status will appear if the process was successful.



```
                        Start Data Monitor (STRDTAMON)

 Type choices, press Enter.

 From . . . . . . . . . . . . . .   *NOW_____     *NOW, *LASTTIME, *DATETIME...
























                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 21 – Starting monitors

Note
For the monitor to start, the journal associated to the file to monitor must be active (status active). If the journal is not active an error appears, the monitor for this journal will not start and Monitor Status in Work with monitored files becomes *JRNINACTI.

Default values are the values shown when using F10 – Start. They are also used for automatic starts after an IPL or after T4DATAMON subsystem start.

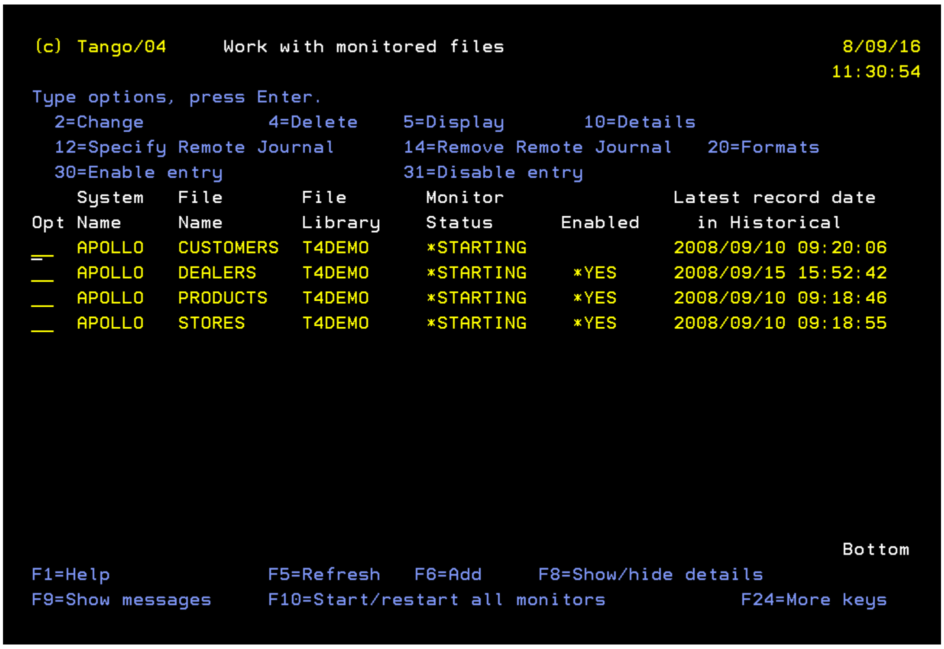
You can determine default values for from what point you want Data Monitor to start processing journal entries by using option **13** from the main Data Monitor menu (T4DATAMON/DTM_MENU). Another way to configure the starting point is to use the following command:

```
T4DATAMON/CHGDTAMON
```

Typically, the first time you run Data Monitor after installing the product, you will use the value *NOW to start processing journal entries from the current moment.

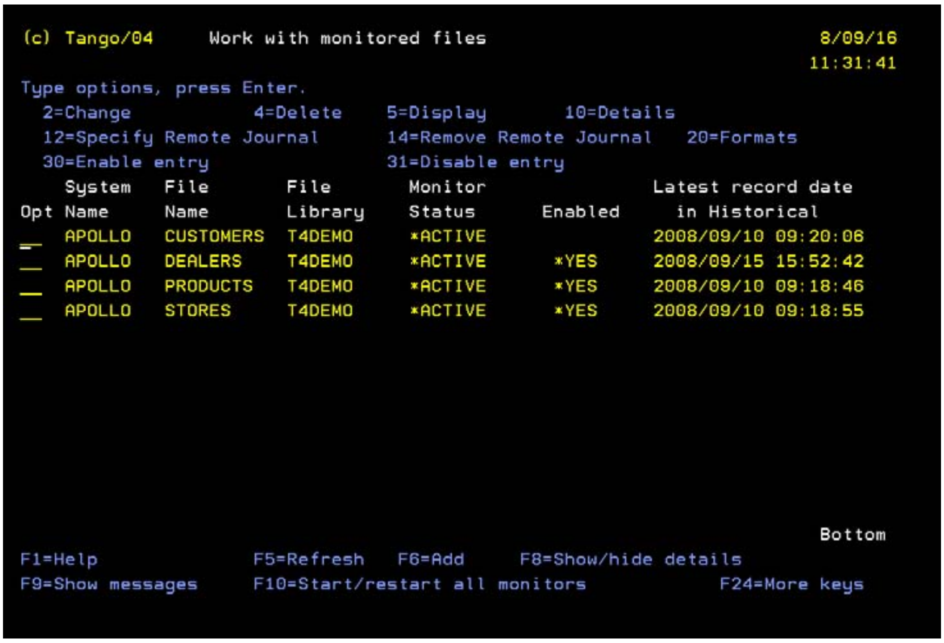
For normal operations once the monitor is up and running, you will set the default value to *LASTTIME. With this setting Data Monitor will start processing journal entries from the last processed entry in the journal. For example after an IPL, Data Monitor will start processing the first journal entry that has not yet been processed.

Only in special cases, for example when re-processing journal entries (*see Appendix C: How to Re-Process Entries on page 99*), will you set the value to *DATETIME to set the starting point to a specific date and time.



```
 (c) Tango/04     Work with monitored files                          8/09/16
                                                                    11:30:54
Type options, press Enter.
  2=Change           4=Delete     5=Display       10=Details
  12=Specify Remote Journal       14=Remove Remote Journal    20=Formats
  30=Enable entry                 31=Disable entry
    System   File        File        Monitor                Latest record date
Opt Name     Name        Library     Status       Enabled     in Historical
__  APOLLO   CUSTOMERS   T4DEMO      *STARTING              2008/09/10 09:20:06
__  APOLLO   DEALERS     T4DEMO      *STARTING      *YES    2008/09/15 15:52:42
__  APOLLO   PRODUCTS    T4DEMO      *STARTING      *YES    2008/09/10 09:18:46
__  APOLLO   STORES      T4DEMO      *STARTING      *YES    2008/09/10 09:18:55









                                                                       Bottom
F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors             F24=More keys
```

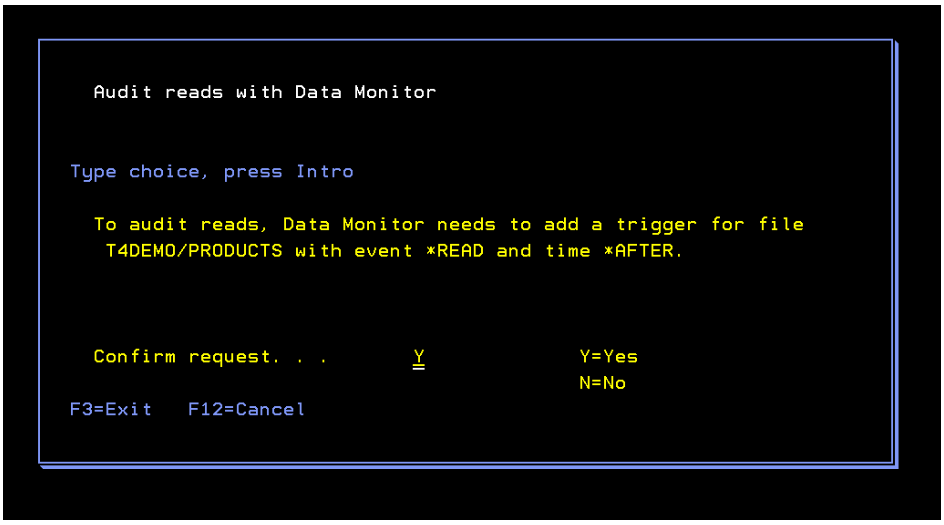
*Figure 22 – After pressing F10 monitor status for enabled entries changes to *STARTING.*



```
 (c) Tango/04     Work with monitored files                          8/09/16
                                                                    11:31:41
Type options, press Enter.
  2=Change           4=Delete     5=Display       10=Details
  12=Specify Remote Journal       14=Remove Remote Journal    20=Formats
  30=Enable entry                 31=Disable entry
    System   File        File        Monitor                Latest record date
Opt Name     Name        Library     Status       Enabled     in Historical
__  APOLLO   CUSTOMERS   T4DEMO      *ACTIVE                2008/09/10 09:20:06
__  APOLLO   DEALERS     T4DEMO      *ACTIVE        *YES    2008/09/15 15:52:42
__  APOLLO   PRODUCTS    T4DEMO      *ACTIVE        *YES    2008/09/10 09:18:46
__  APOLLO   STORES      T4DEMO      *ACTIVE        *YES    2008/09/10 09:18:55









                                                                       Bottom
F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors             F24=More keys
```

*Figure 23 – Monitors that were successfully started will show Monitor Status *ACTIVE.*

To audit reads with Data Monitor a trigger must be added to the file. Therefore when you start a monitor that will also check reads the following screen appears:



```
  Audit reads with Data Monitor

Type choice, press Intro

  To audit reads, Data Monitor needs to add a trigger for file
   T4DEMO/PRODUCTS with event *READ and time *AFTER.


  Confirm request. . .       Y             Y=Yes
                                           N=No
F3=Exit   F12=Cancel
```

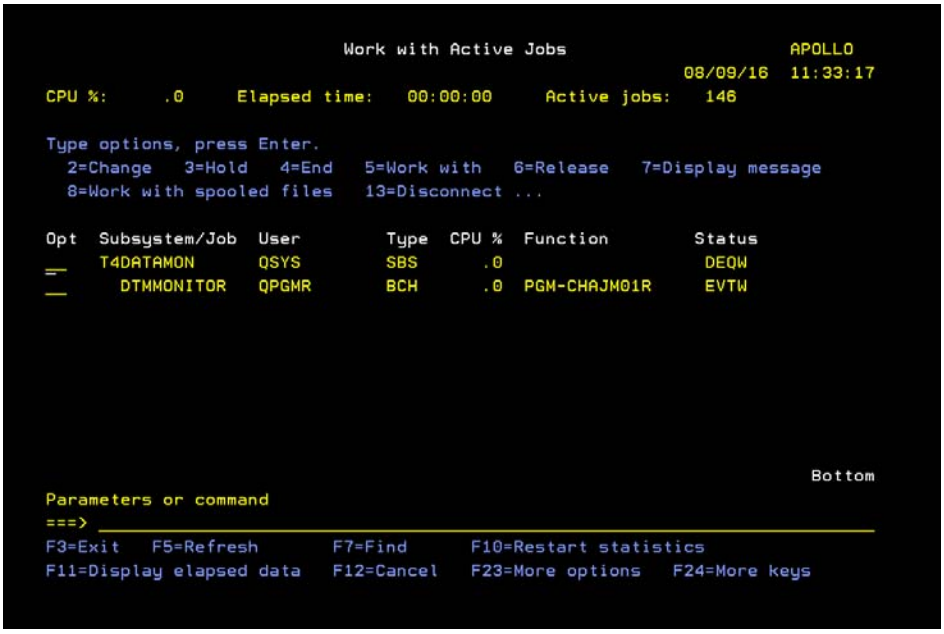
Figure 24 – Confirm request to add triggers for Reads.

If you confirm the request, Data Monitor for iSeries will add the triggers so that you can monitor reads.

3.6.1 The T4DATAMON subsystem

All monitor jobs of the Data Monitor run under the `T4DATAMON` subsystem. A monitor is responsible for processing journal entries of all files that are journaled by the same journal.

Thus, instead of a monitor for each monitored file, there is a monitor for each journal that has an associated monitored file, as shown in the next figure:



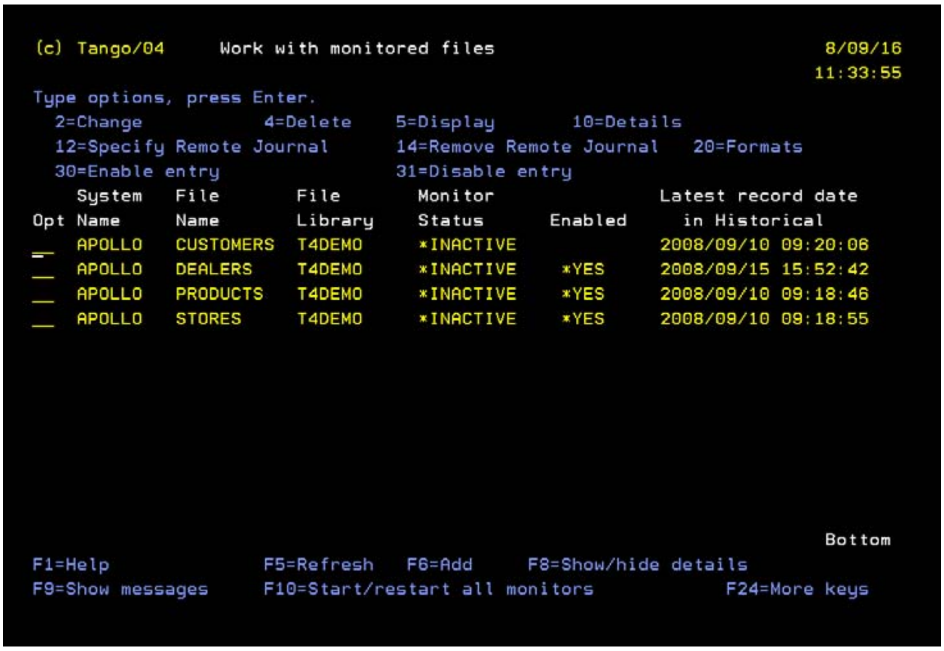
```
                          Work with Active Jobs                    APOLLO
                                                         08/09/16  11:33:17
CPU %:      .0     Elapsed time:   00:00:00     Active jobs:    146

Type options, press Enter.
  2=Change    3=Hold    4=End    5=Work with    6=Release    7=Display message
  8=Work with spooled files     13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function          Status
 __  T4DATAMON      QSYS        SBS      .0                     DEQW
 __    DTMMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R       EVTW







                                                                      Bottom
Parameters or command
===> __________________________________________________________________________
F3=Exit   F5=Refresh        F7=Find      F10=Restart statistics
F11=Display elapsed data    F12=Cancel   F23=More options   F24=More keys
```

Figure 25 – Jobs for each journal, whose associated files we are monitoring.

To find what monitor is related to a given file *see section 3.2 - Displaying File Details in Data Monitor on page 11*.

3.7 Ending Monitors

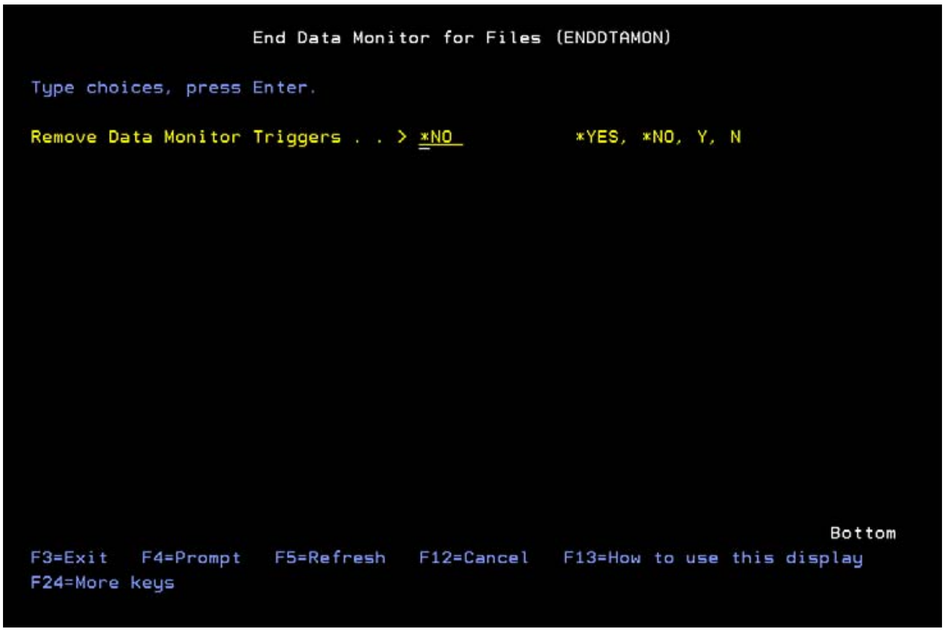
You can use F11 – End all monitors, from the Work with monitored files screen to end all monitors associated to monitored files. If the process was successful, the Monitor status will appear as `*INACTIVE`.



```
 (c) Tango/04      Work with monitored files                                 8/09/16
                                                                            11:33:55
 Type options, press Enter.
   2=Change             4=Delete     5=Display        10=Details
   12=Specify Remote Journal         14=Remove Remote Journal    20=Formats
   30=Enable entry                   31=Disable entry
     System    File        File        Monitor                 Latest record date
 Opt Name      Name        Library     Status        Enabled     in Historical
 __  APOLLO    CUSTOMERS   T4DEMO      *INACTIVE               2008/09/10 09:20:06
 __  APOLLO    DEALERS     T4DEMO      *INACTIVE       *YES    2008/09/15 15:52:42
 __  APOLLO    PRODUCTS    T4DEMO      *INACTIVE       *YES    2008/09/10 09:18:46
 __  APOLLO    STORES      T4DEMO      *INACTIVE       *YES    2008/09/10 09:18:55











                                                                              Bottom
 F1=Help                F5=Refresh    F6=Add      F8=Show/hide details
 F9=Show messages       F10=Start/restart all monitors              F24=More keys
```

*Figure 26 – Once monitors have been ended using F11 the Monitor Status changes to *INACTIVE.*

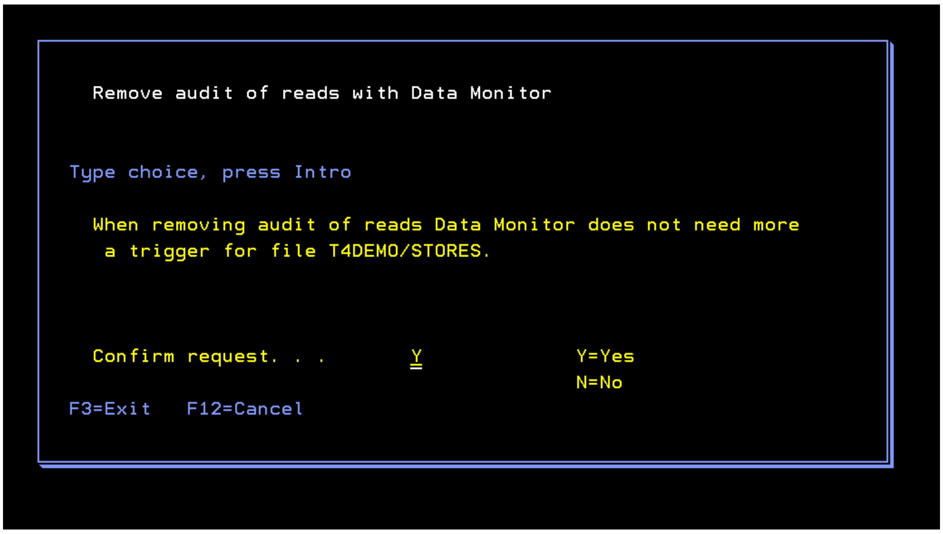
When ending a monitor that monitors *Reads* you will be asked whether you want to remove the triggers.



```
                      End Data Monitor for Files (ENDDTAMON)

 Type choices, press Enter.

 Remove Data Monitor Triggers . . > *NO_          *YES, *NO, Y, N

















                                                                              Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 27 – Remover triggers for Reads when ending a monitor

If you confirm the request, Data Monitor for iSeries will remove the triggers so that messages for Reads are no longer sent to the database. Data Monitor for iSeries will ask for each file with triggers whether or not to remove them, as shown in the next figure.



```
   Remove audit of reads with Data Monitor


 Type choice, press Intro

   When removing audit of reads Data Monitor does not need more
    a trigger for file T4DEMO/STORES.



   Confirm request. . .       Y             Y=Yes
                                            N=No
 F3=Exit   F12=Cancel
```

Figure 28 – Option to remove triggers file by file.

In certain cases you may want to end the monitor, but continue to audit the Reads that occur while the monitor is inactive. In this case do not remove the trigger and the next time you start the monitor all the messages that were sent in the interim will be processed by the monitor.

3.8 Monitoring Files on Remote Systems

You can add files from remote systems to be monitored by Data Monitor, because OS/400 (or i5/OS) supports remote journals. Basically remote journals are replications of journals and journal receivers from other systems in the system where Data Monitor resides. If you want to implement this option you should have a basic knowledge of remote journaling. You can find detailed information at IBM iSeries(TM) Information Center.

Remote journaling can also be used as a way to move Data Monitor processing off of your production box. You can create journals, remote journals and receivers on other iSeries machines and process transactions made on your production box from these other iSeries machines. This way Data Monitor will have near ZERO PERFORMANCE IMPACT on your production machine.

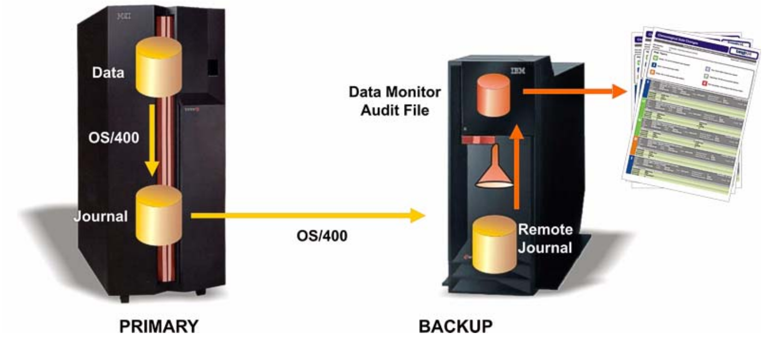
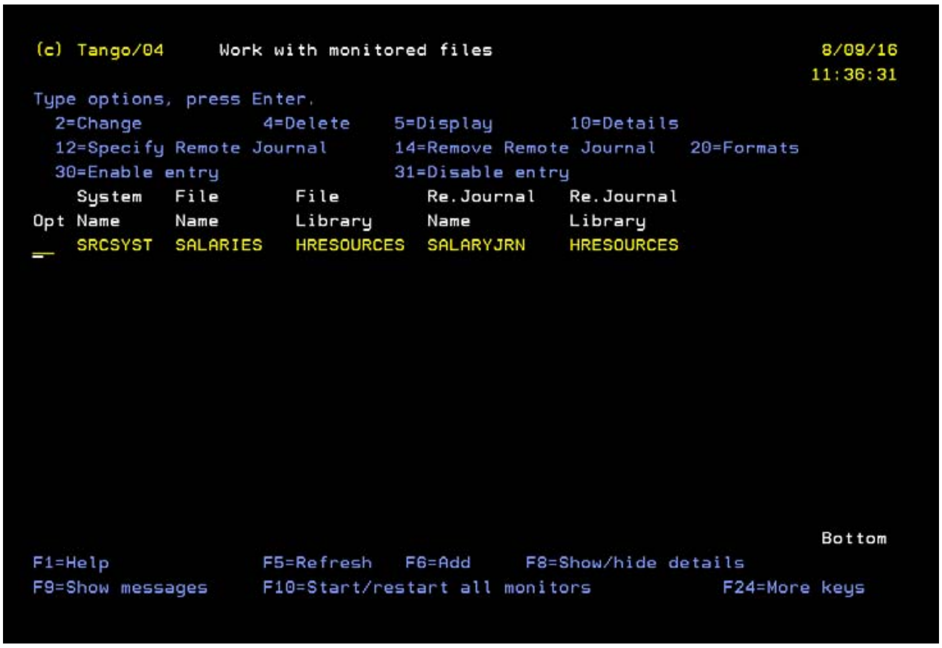


Figure 29 – Near zero impact on performance of your production machine, when using Data Monitor with remote journaling

3.8.1 Configuring remote files for changes

Assuming that you already have a remote journal where transactions for a given file are audited, first add the file. From the work with monitored files screen use F6 to create the new file. Enter the File name and library, and press F10 to view additional parameters. Enter the *source system name*.

Next, add or remove a remote journal of a file in Data Monitor from the Work with files screen using option **12** to add a remote journal, or option **14** to remove a remote journal. Simply add the *journal* and *library name* of the remote journal.



```
(c) Tango/04     Work with monitored files                                8/09/16
                                                                         11:36:31
Type options, press Enter.
  2=Change           4=Delete    5=Display       10=Details
  12=Specify Remote Journal      14=Remove Remote Journal   20=Formats
  30=Enable entry                31=Disable entry
    System   File        File        Re.Journal    Re.Journal
Opt Name     Name        Library     Name          Library
__  SRCSYST  SALARIES    HRESOURCES  SALARYJRN     HRESOURCES






                                                                         Bottom
F1=Help              F5=Refresh   F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors           F24=More keys
```

Figure 30 – File with remote journal in view 2 (F13)

It is important to understand that although the remote file you are adding does not necessarily need to exist on your local system (where you have Data Monitor installed), the files for parameters *Initial format (from file)* and *Initial key list* must physically exist on your local system, because Data Monitor uses these to store the format (or schema) required for analyzing the remote file transactions.

Note that if you have recently added other formats for this remote file you must also have these format files on your local system.

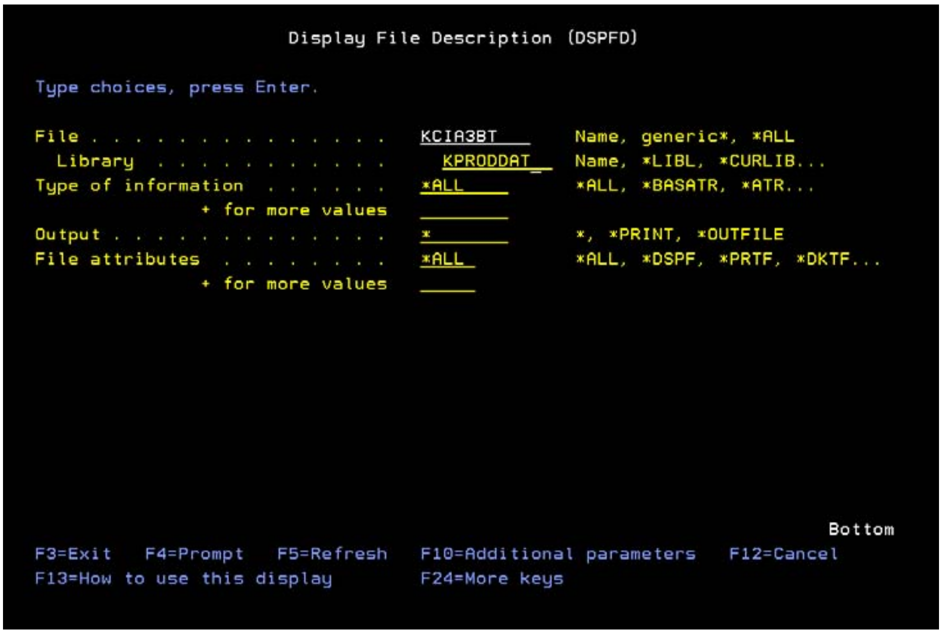
Once you have added all the remote files you want to monitor, there is one additional step. Because Data Monitor reads from a remote journal and the remote file does not exist on your local system, it is not possible for Data Monitor to retrieve the name of the remote journal to read transactions from, nor is it possible to know if the file is being journaled in the source system.

First you must manually check whether the file is being journaled. Then you must tell Data Monitor what remote journal to read by entering the appropriate journal in option **12** of the Work with monitored files menu.

How to enter the appropriate journal:

Step 1. Go to the source system and use the `DSPFD` command.

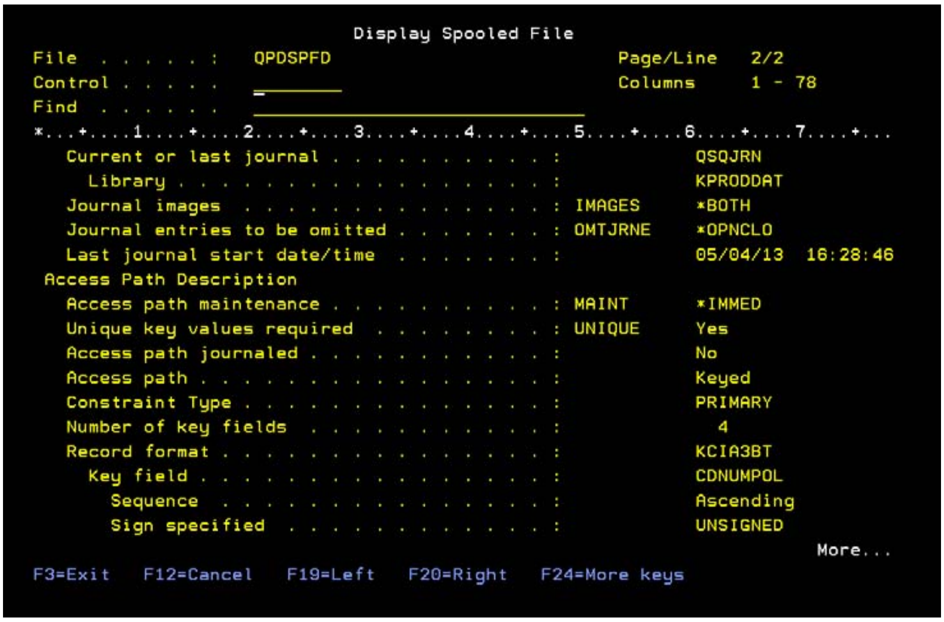
Step 2. Enter the *file name* you are interested in and press ENTER



```
                       Display File Description (DSPFD)

Type choices, press Enter.

File . . . . . . . . . . . . . . > KCIA3BT       Name, generic*, *ALL
  Library  . . . . . . . . . . .     KPRODDAT    Name, *LIBL, *CURLIB...
Type of information  . . . . . .   *ALL          *ALL, *BASATR, *ATR...
               + for more values
Output . . . . . . . . . . . . .   *             *, *PRINT, *OUTFILE
File attributes  . . . . . . . .   *ALL          *ALL, *DSPF, *PRTF, *DKTF...
               + for more values

                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 31 – Enter the file name

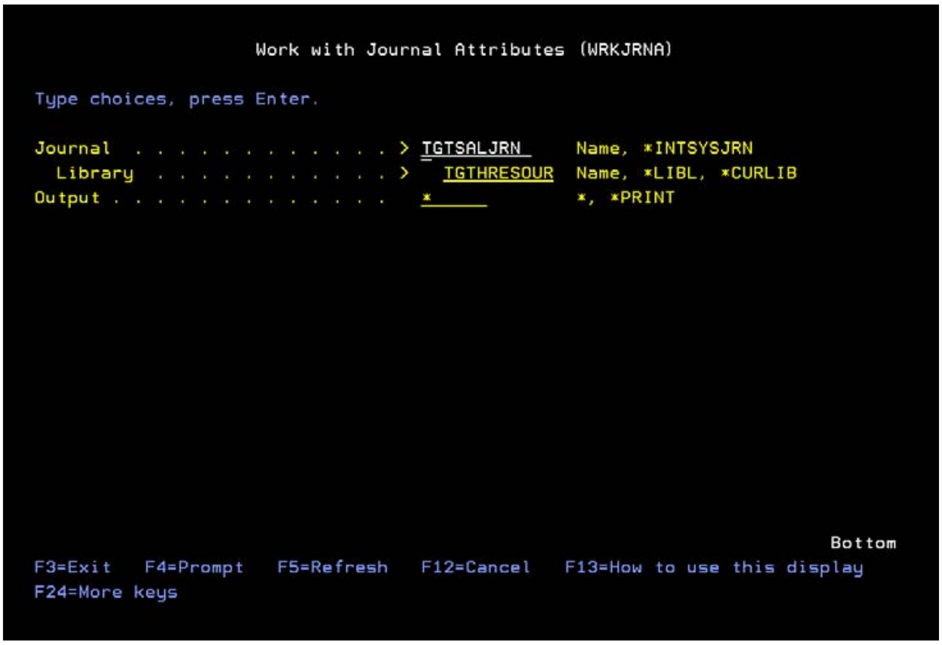
Step 3. In the resulting screen the *Current or last journal* field shows the name of the journal in the source system. In this example the journal name is `QSQJRN`. This is the local journal.



```
                             Display Spooled File
File  . . . . . :   QPDSPFD                         Page/Line   2/2
Control . . . . .                                   Columns     1 - 78
Find  . . . . . .
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
   Current or last journal . . . . . . . . . . :               QSQJRN
     Library . . . . . . . . . . . . . . . . . :               KPRODDAT
   Journal images  . . . . . . . . . . . . . . : IMAGES        *BOTH
   Journal entries to be omitted . . . . . . . : OMTJRNE       *OPNCLO
   Last journal start date/time  . . . . . . . :               05/04/13  16:28:46
 Access Path Description
   Access path maintenance . . . . . . . . . . : MAINT         *IMMED
   Unique key values required  . . . . . . . . : UNIQUE        Yes
   Access path journaled . . . . . . . . . . . :               No
   Access path . . . . . . . . . . . . . . . . :               Keyed
   Constraint Type . . . . . . . . . . . . . . :               PRIMARY
   Number of key fields  . . . . . . . . . . . :                 4
   Record format . . . . . . . . . . . . . . . :               KCIA3BT
     Key field . . . . . . . . . . . . . . . . :               CDNUMPOL
       Sequence  . . . . . . . . . . . . . . . :               Ascending
       Sign specified  . . . . . . . . . . . . :               UNSIGNED
                                                                        More...
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

Figure 32 – The name of the journal in the source system is displayed in the Current or last journal field

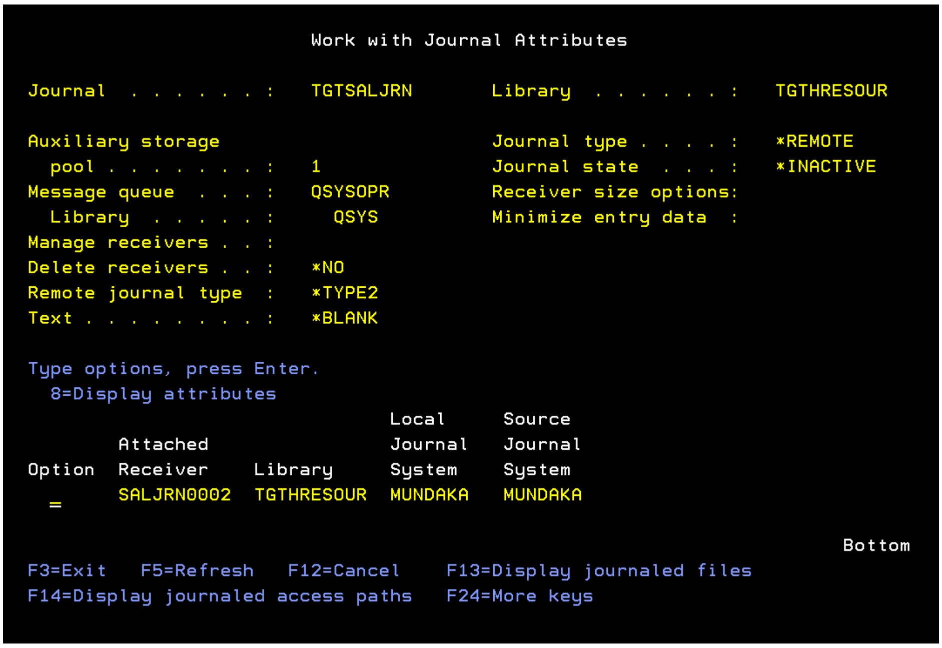
Step 4. Next use the `WRKJRNA` command from the source system to check the remote journal associated with the journal in the source system.



```
                   Work with Journal Attributes (WRKJRNA)

 Type choices, press Enter.

 Journal . . . . . . . . . . . . > TGTSALJRN     Name, *INTSYSJRN
   Library . . . . . . . . . . . >   TGTHRESOUR  Name, *LIBL, *CURLIB
 Output . . . . . . . . . . . . .   *            *, *PRINT

                                                                    Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 33 – Work with Journal attributes

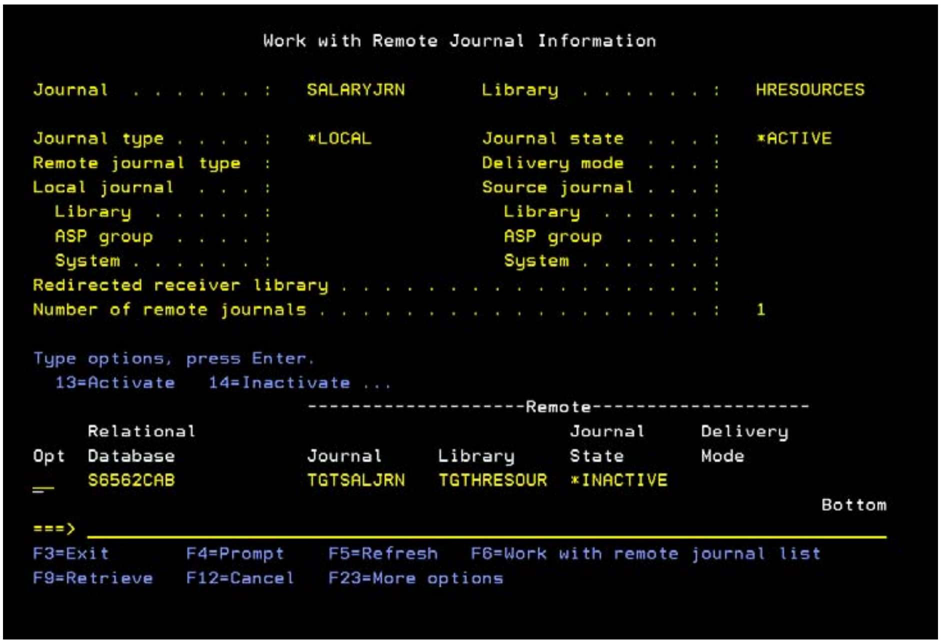
Step 5. Enter the *journal name* and press ENTER. The following screen appears:



```
                          Work with Journal Attributes

 Journal . . . . . . :   TGTSALJRN       Library . . . . . . :   TGTHRESOUR

 Auxiliary storage                       Journal type . . . . :   *REMOTE
   pool . . . . . . . :   1              Journal state . . . :   *INACTIVE
 Message queue . . . :   QSYSOPR         Receiver size options:
   Library . . . . . :     QSYS          Minimize entry data  :
 Manage receivers . . :
 Delete receivers . . :   *NO
 Remote journal type  :   *TYPE2
 Text . . . . . . . . :   *BLANK

 Type options, press Enter.
   8=Display attributes
                                    Local     Source
         Attached                   Journal   Journal
 Option  Receiver    Library        System    System
         SALJRN0002  TGTHRESOUR     MUNDAKA   MUNDAKA
   _
                                                                       Bottom
 F3=Exit   F5=Refresh   F12=Cancel    F13=Display journaled files
 F14=Display journaled access paths   F24=More keys
```

Figure 34 – Journal details

Step 6. Press F16 – Work with remote journal information – to display the name of the associated remote journal, as shown in the next image.



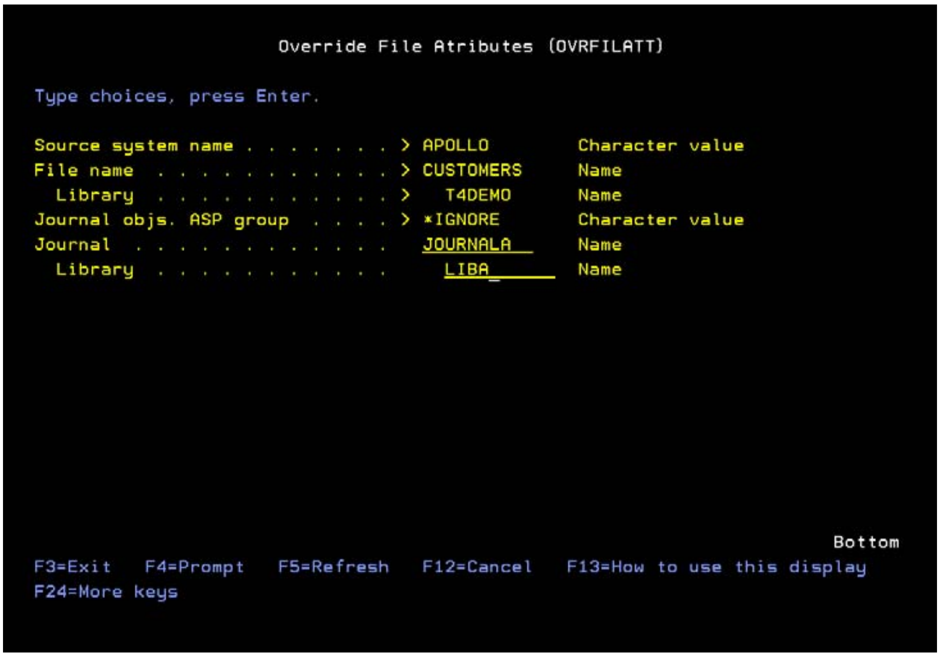
```
                     Work with Remote Journal Information

 Journal  . . . . . . :   SALARYJRN       Library  . . . . . . :   HRESOURCES

 Journal type . . . . :   *LOCAL          Journal state  . . . :   *ACTIVE
 Remote journal type  :                   Delivery mode  . . . :
 Local journal  . . . :                   Source journal . . . :
   Library  . . . . . :                     Library  . . . . . :
   ASP group  . . . . :                     ASP group  . . . . :
   System . . . . . . :                     System . . . . . . :
 Redirected receiver library . . . . . . . . . . . . . . . . . :
 Number of remote journals . . . . . . . . . . . . . . . . . . :   1

 Type options, press Enter.
   13=Activate   14=Inactivate ...
                            --------------------Remote--------------------
      Relational                                    Journal     Delivery
 Opt  Database              Journal     Library     State       Mode
 _    S6562CAB              TGTSALJRN   TGTHRESOUR  *INACTIVE
                                                                      Bottom
 ===> __________________________________________________________________________
 F3=Exit       F4=Prompt    F5=Refresh   F6=Work with remote journal list
 F9=Retrieve   F12=Cancel   F23=More options
```

Figure 35 – Work with remote journal information

Step 7. This is the *remote journal name* and *library* you have to set in option **12** of the Work with monitored files menu.

An example:

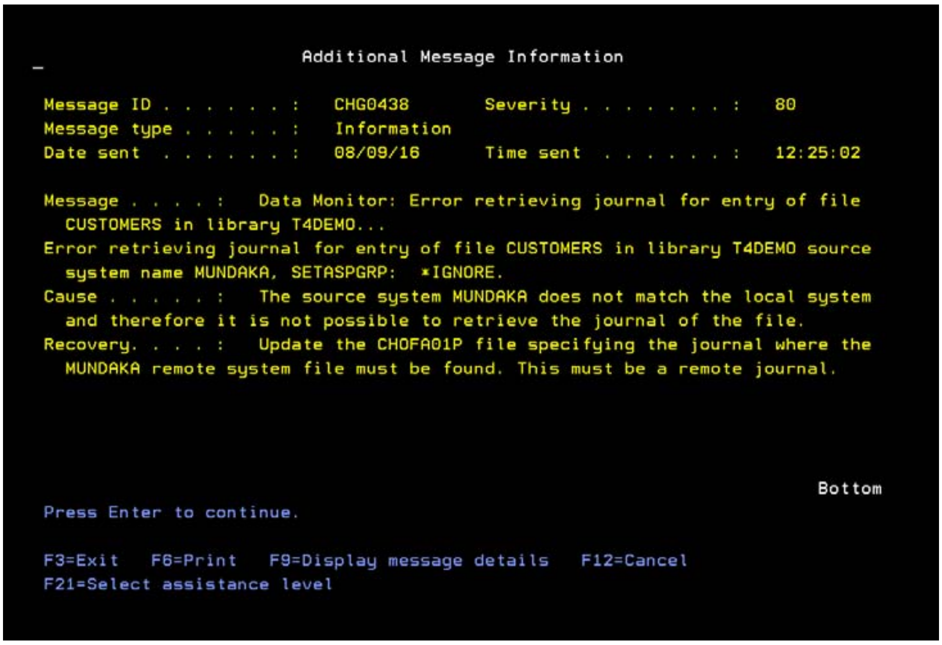
Imagine you are running Data Monitor on system APOLLO and you have added an entry for file FILEA in library LIBA for remote system DIONISO. You know that the remote journal that receives information about transactions in this file is located on system APOLLO and is called JOURNALA in library LIBA. Add the following entry to option **12** in work with monitored files menu:



```
                      Override File Atributes (OVRFILATT)

 Type choices, press Enter.

 Source system name . . . . . . . > APOLLO         Character value
 File name  . . . . . . . . . . . > CUSTOMERS      Name
   Library  . . . . . . . . . . . >   T4DEMO       Name
 Journal objs. ASP group  . . . . > *IGNORE        Character value
 Journal  . . . . . . . . . . . .   JOURNALA__     Name
   Library  . . . . . . . . . . .     LIBA______   Name









                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 36 – Enter the details to override file attributes

If you make any mistakes when adding a remote file to Data Monitor a message will appear when you press F10 to start/restart monitors. The message will inform you that the journal for this file could not be retrieved and the monitor could not be started.



```
_                        Additional Message Information

 Message ID . . . . . . :   CHG0438       Severity . . . . . . . :   80
 Message type . . . . . :   Information
 Date sent  . . . . . . :   08/09/16      Time sent  . . . . . . :   12:25:02

 Message . . . . :   Data Monitor: Error retrieving journal for entry of file
   CUSTOMERS in library T4DEMO...
 Error retrieving journal for entry of file CUSTOMERS in library T4DEMO source
   system name MUNDAKA, SETASPGRP:  *IGNORE.
 Cause . . . . . :   The source system MUNDAKA does not match the local system
   and therefore it is not possible to retrieve the journal of the file.
 Recovery. . . . :   Update the CHOFA01P file specifying the journal where the
   MUNDAKA remote system file must be found. This must be a remote journal.



                                                                         Bottom
 Press Enter to continue.

 F3=Exit   F6=Print   F9=Display message details   F12=Cancel
 F21=Select assistance level
```

Figure 37 – Error message received if the journal could not be retrieved

3.8.2 Configuring remote files for changes and reads - Example

We will explain the process using an example. In our example we want to monitor the file `SALARIES` in library `HRESOURCES` from a remote machine (target system). In this example we assume that remote files are not journaled at all; neither at the *remote system* (also called *source system*), nor at *target system*.

To monitor a file on a remote system it is mandatory that the remote file is journaled. If the file is not yet journaled create the journal as follows:

Journaling a file

Step 1. Create the journal receiver using for example the following command on the source system[4]:

```
CRTJRNRCV JRNRCV(HRESOURCES/SALJRN0001)
```

Step 2. Create the journal using the following command on source system:

```
CRTJRN   JRN(HRESOURCES/SALARYJRN)   JRNRCV(HRESOURCES/SALJRN0001)
MNGRCV(*SYSTEM) FIXLENDTA(*JOB *USR *PGM *PGMLIB *RMTADR)
```

Step 3. Use this command to start physical file journal on the source system:

```
STRJRNPF    FILE(HRESOURCES/SALARIES)    JRN(HRESOURCES/SALARYJRN)
IMAGES(*BOTH)
```

Step 4. Create the remote journal on the target system where Data Monitor is installed.

- First create a library on the target system for creating the remote journal, using the following command:

```
CRTLIB LIB(TGTHRESOUR)
```

- Next add the remote journal from the source system.

```
ADDRMTJRN       RDB(TGTSYSTEM)       SRCJRN(HRESOURCES/SALARYJRN)
TGTJRN(TGTHRESOUR/TGTSALJRN)RMTRCVLIB(TGTHRESOUR)RMTJRNTYPE
(*TYPE2)
```

[4] Source system is the system on which the file to be monitored is located. Target system is the system from which you will be monitoring the journaled file.

Step 5. Finally activate and synchronize the remote journal using the following command from the source system:

```
CHGRMTJRN        RDB(TGTSYSTEM)        SRCJRN(HRESOURCES/SALARYJRN)
TGTJRN(TGTHRESOUR/TGTSALJRN)JRNSTATE(*ACTIVE)DELIVERY(*SYNC)
```

Adding triggers (only for monitoring read transactions)

Manually add the trigger to the file on the source system. With the following command:

```
ADDPFTRG    FILE(HRESOURCES/SALARIES)    TRGTIME(*AFTER)    TRGEVENT(*READ)
PGM(T4DATAMON/CHTRG01R)TRG(T4HRESOURCESSALARIES)
```

Where the naming convention of the trigger is:

```
T4 + Library + Filename, in our example: T4HRESOURCESSALARIES
```

Note

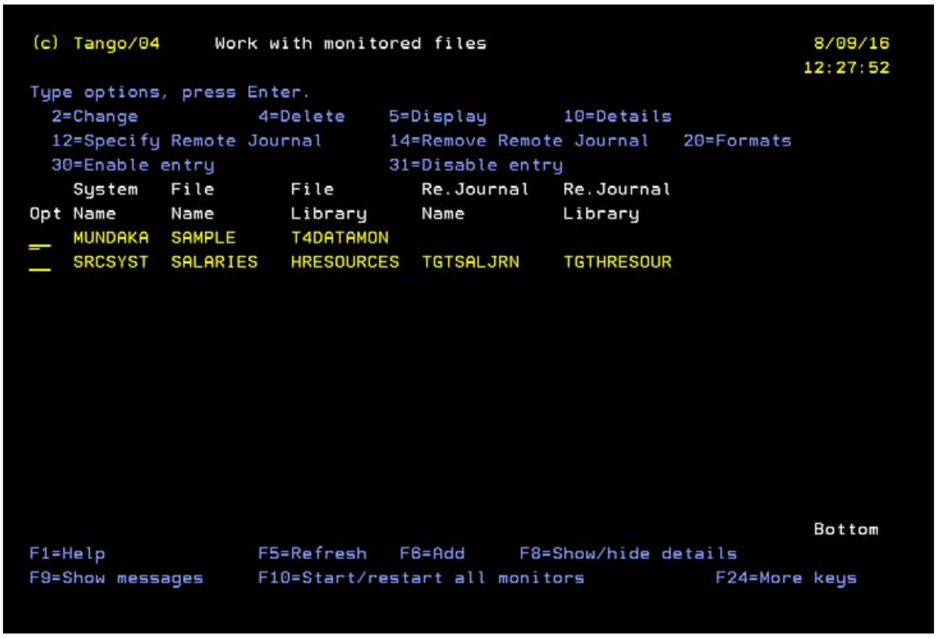
If you need to add triggers to monitor reads you will also need to install Data Monitor on this source system, because you need a number of programs included in the product. You won't be running these installations actively and therefore do not need to configure the product or enter the product activation keys.

Configuring the target system

From the target system you must configure the `HRESOURCE SALARIES` file.

Use F6 to add the `SALARIES` file. From the Add Data Monitor File screen use F10 (additional parameters) and change the source system name.

Then use option **12** to add a remote journal for this file. Once the file is correctly configured you will see the following screen.



```
 (c) Tango/04     Work with monitored files                          8/09/16
                                                                    12:27:52
Type options, press Enter.
  2=Change            4=Delete     5=Display       10=Details
  12=Specify Remote Journal        14=Remove Remote Journal    20=Formats
  30=Enable entry                  31=Disable entry
    System   File        File        Re.Journal    Re.Journal
Opt Name     Name        Library     Name          Library
__  MUNDAKA  SAMPLE      T4DATAMON
__  SRCSYST  SALARIES    HRESOURCES  TGTSALJRN     TGTHRESOUR

                                                             Bottom
F1=Help              F5=Refresh   F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors            F24=More keys
```

Figure 38 – Remote journal added

Use option **2** to change the flag *audit read values* to `*YES` for this file.

Finally press F10 to start the monitor.

Once the monitor has been started you will notice that the field latest record date in historical is updated for every change or read operation.

3.8.3 Monitoring Data Monitor databases

In certain cases you may wish to monitor Data Monitors own transaction databases. However you will notice that if you simply enter a Data Monitor database name and the library it will not allow you to complete the entry, nor will you be allowed to change an existing entry so that it monitors a local Data Monitor database.

As you have learned Data Monitor stores events for changes to the journaled files it monitors in the transaction databases. If you were to monitor Data Monitor's transaction database, Data Monitor would also add a new record to this transaction database for every record added or changed in the transaction database, thus creating an endless loop.

To avoid this situation Data Monitor does not allow you to add a configuration entry for a local Data Monitor database. If you want to monitor a Data Monitor transaction database you must do so from a remote machine, using remote journaling for details *see section 3.8.1 - Configuring remote files for changes on page 35*.

Note
When monitoring Data Monitor databases from remote machines, you must continue to take care not to create another endless loop!

At the risk of stating the obvious we'd like to elaborate on this with two examples:

Example 1

If you have two separate systems (system A and system B) each running Data Monitor, you could use system B to monitor the Data Monitor transaction databases on system A.

Once you have set up this configuration on system B, beware NOT to add an entry on system A to monitor the Data Monitor transaction databases on system B, as this would also create an endless loop.

Example 2

If you have a network of several separate systems (let's say system A, B, C, and D) you can use system D to monitor system C, System C to monitor System B, System B to monitor System A. Again take extreme care not to close the circle by configuring System A to monitor System D or you will be creating another endless loop.

Chapter 4
Configuring and Starting Data Monitor to monitor Sensitive Users

Data Monitor for iSeries can be configured to monitor sensitive files and sensitive users. Using Data monitor for iSeries allows you to track all actions for a particular user, or you can narrow down your audit to monitor transactions in particular journals or files of a particular user.

When users read or make changes to journaled files, information regarding the transaction is stored in a journal. A single journal may store information regarding user transactions from multiple journaled files. Data Monitor then checks all these journals to find information regarding transactions of sensitive users to monitor.

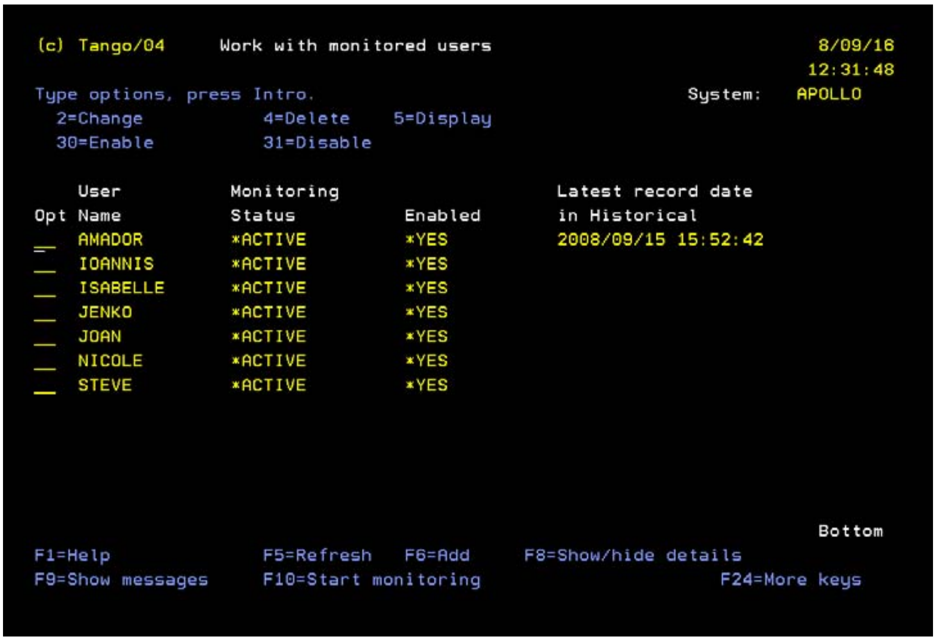
As you may well imagine Data Monitor potentially needs to trawl through an even larger set of data than when monitoring sensitive files alone. Therefore, we recommend you carefully select only those sensitive users that are of utmost importance to the security of your data. Furthermore, check that you select only those journals that contain sensitive data.

When monitoring sensitive users you can find transactions of any file in your system. Then there are no formats to process transactions as when you manually select sensitive files. That is why Data Monitor takes care of loading a format when it finds the first journal entry for a specific file (Load Formats on the fly function). Where possible, we suggest you reuse the configured file formats (default setting in Data Monitor), so that Data Monitor will reuse your format configuration for sensitive files when monitoring sensitive users. In addition, it is a good idea to regularly clean up your Data Monitor files.

This chapter describes Data Monitor for iSeries Sensitive Users. The first sections discuss the screens you will use to check your Data Monitor configuration and the basic variables displayed in these screens. The remaining sections discuss how to add new entries and modify existing configuration entries.

4.1 Viewing Data Monitor Users Configuration

From the General Settings screen (menu `DTM_CONFIG`), use option **2 -** Work with monitored users to see all the configuration entries for the Data Monitor.



```
 (c) Tango/04     Work with monitored users                          8/09/16
                                                                     12:31:48
 Type options, press Intro.                               System:   APOLLO
   2=Change             4=Delete     5=Display
   30=Enable            31=Disable

     User          Monitoring                   Latest record date
 Opt Name          Status          Enabled      in Historical
 _   AMADOR        *ACTIVE         *YES         2008/09/15 15:52:42
 _   IOANNIS       *ACTIVE         *YES
 _   ISABELLE      *ACTIVE         *YES
 _   JENKO         *ACTIVE         *YES
 _   JOAN          *ACTIVE         *YES
 _   NICOLE        *ACTIVE         *YES
 _   STEVE         *ACTIVE         *YES

                                                                    Bottom
 F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
 F9=Show messages     F10=Start monitoring                     F24=More keys
```

Figure 39 – Sensitive users to monitor

The first column, *User Name*, identifies the sensitive users that have been added to the Data Monitor configuration. You can enter a specific user name (`JOAN`), or simply enter `*ALL` to monitor all users.

Tip

Try not to repeat users. For example, if you choose to use user `*ALL`, do not add more users – this will only lead Data Monitor to monitor the same user twice.

Monitor Status indicates the job status of the job dedicated to monitoring the configured user.

For the Data Monitor to monitor a particular entry the entry must be *enabled*. An entry that is not enabled will not be monitored when the Data Monitor is restarted. For more information *see section 5.4 - Starting/ Restarting Monitors on page 57.*

Latest record date in Historical indicates the date and time of the last record stored in the historical transactions database for the user in this configuration entry.

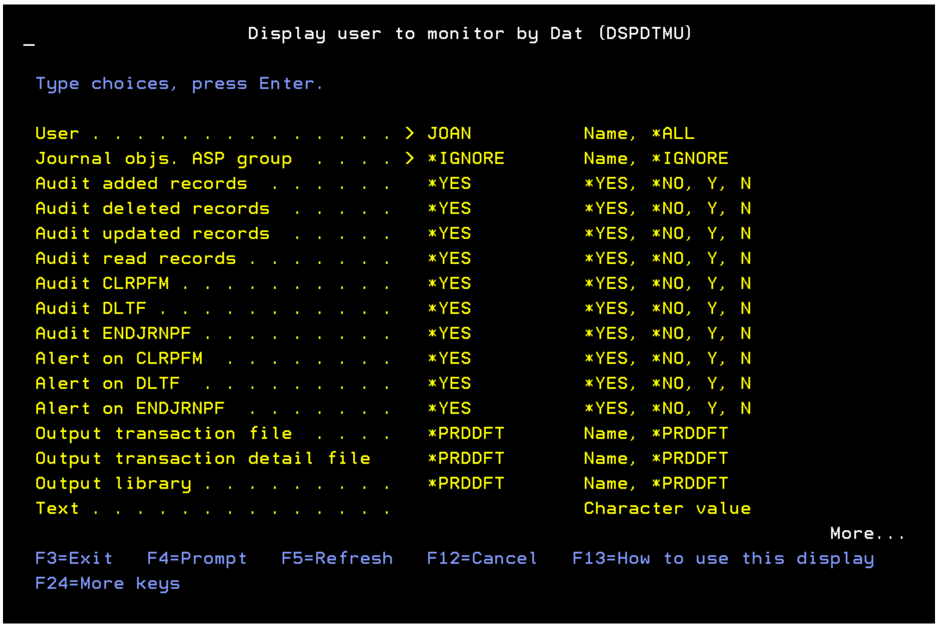
Of course, if this is the first time you are using the product this screen will be empty. You can add new configuration entries using F6. *See section 5.3 - Adding a User to Data Monitor on page 54* for details.

4.2 Displaying User Settings in Data Monitor

From the Work with monitored users screen, use option **5** to display the settings of a monitored user. Another way to display the file details is to enter the following command:

```
T4DATAMON/DSPDTMU and press F4.
```

Here enter the user you want to view.



```
_                     Display user to monitor by Dat (DSPDTMU)

 Type choices, press Enter.

 User . . . . . . . . . . . . . . > JOAN          Name, *ALL
 Journal objs. ASP group  . . . . > *IGNORE       Name, *IGNORE
 Audit added records  . . . . . .   *YES          *YES, *NO, Y, N
 Audit deleted records  . . . . .   *YES          *YES, *NO, Y, N
 Audit updated records  . . . . .   *YES          *YES, *NO, Y, N
 Audit read records . . . . . . .   *YES          *YES, *NO, Y, N
 Audit CLRPFM . . . . . . . . . .   *YES          *YES, *NO, Y, N
 Audit DLTF . . . . . . . . . . .   *YES          *YES, *NO, Y, N
 Audit ENDJRNPF . . . . . . . . .   *YES          *YES, *NO, Y, N
 Alert on CLRPFM  . . . . . . . .   *YES          *YES, *NO, Y, N
 Alert on DLTF  . . . . . . . . .   *YES          *YES, *NO, Y, N
 Alert on ENDJRNPF  . . . . . . .   *YES          *YES, *NO, Y, N
 Output transaction file  . . . .   *PRDDFT       Name, *PRDDFT
 Output transaction detail file     *PRDDFT       Name, *PRDDFT
 Output library . . . . . . . . .   *PRDDFT       Name, *PRDDFT
 Text . . . . . . . . . . . . . .                 Character value
                                                                    More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 40 – Display user settings

The user settings screen shows the current data for this configuration entry. Note that there are several screens of file details. Press PgDn to see more file details.

Here follows a brief description of the fields available on the first screen of file details.

User identifies the sensitive user that is being monitored.

Journal Obj. ASP group (only for V5R2M0 and later): This value is used when searching journal and journal receiver objects where file is journaled. If you were not aware of ASP groups, or do not work with ASP groups in your system, ignore this parameter and set it to the default special value `*IGNORE`. If you work with ASP groups enter the ASP group name here.

The next set of variables all start with *Audit* and indicates what types of transactions or commands will be monitored. If you enter `*YES` for any of these options, they will be available in the Historical Transaction Log and can be used in Reports.

Alerts are important events that affect monitoring sensitive users. For example an alert can be sent to VISUAL Message Center when a user ends journaling for one or more files. Indicate here what alerts will be sent to VISUAL Message Center.

Note
The alerts configuration is independent of the file auditing configuration above.

Monitor status: Current monitor status. It indicates the job status of the job dedicated to monitor the file configured in Data Monitor. The following valid values are possible:

Status	Description
*ACTIVE	The monitor is active and in a correct state.
*JOBQ	The monitor is in the queue but has not yet been started. This value is platform-dependent. iSeries uses queues when starting processes.
*OUTQ	The monitor is not active. The previous session ended correctly but left a log.
*INITFAIL	An error occurred while the monitor was in `*STARTING` status.

Status	Description
*STARTING	The monitor is still in the initialization phase.
*INACTIVE	This monitor is not started and there is no evidence that this monitor has been started in the past.
*ENDING	The monitor is being ended from the interface.
*INERROR	The monitor is active, but not responding.
*FAILED	The monitor ended due to an error that occurred while in `*ACTIVE` state.
*RESTART	The monitor is being restarted from the interface.
*JRNINACTI	The journal is temporarily inactive. The monitor is active and waiting for the journal to be activated so that it can return to analyzing the operations.
N/A	An error occurred while trying to retrieve the monitor status.

Press PgDn to display more file details, including:

Text: Here you can enter a description of the user. This description appears in the Work with sensitive users screen when you press F8 (show/hide details).

Status: the user can enable or disable the configuration entry. The only valid values in this field are `*ENABLED` or `*DISABLED`. When the user executes function F10 (start and reinitiate all monitors) only the enabled entries will be started or reinitialized.

4.3 Adding a User to Data Monitor

Now that you know where to find information about the current configuration we will discuss how to create new entries and modify existing entries.

Although Data Monitor reads transactions from journals, it can also work at user level, not only at journal level. Data Monitor only processes transactions for the sensitive users that are specified, and not those of the entire journal.

Note

Data Monitor does not journal physical files. Files must be journaled in advance using the `STRJRNPF` command, with parameter IMAGES set to (`*BOTH`). Data Monitor checks this requirement when starting the monitor.

Data Monitor will only monitor a journal if it stores user data. Make sure the journals in Data Monitor's journal list specify the value `*USR` in the parameter `FIXLENDTA`. For details *see Appendix K: Frequently Asked Questions on page 147*.

Data Monitor can only monitor Reads by sensitive users if the sensitive file has a trigger. For details regarding triggers and reads *see Chapter 3 - Configuring and Starting Data Monitor to monitor Sensitive Files on page 10* .

`MINENTDTA(*FILE)` is not supported in Journals. To check the value of parameter `MINENTDTA` use command `WRKJRNA` on the journal.

The journal for the physical file needs to have parameter `MINENTDTA` (minimized entry-specific data) set to a value different from `*FILE`. IBM iSeries Information Center explains this note as follows:

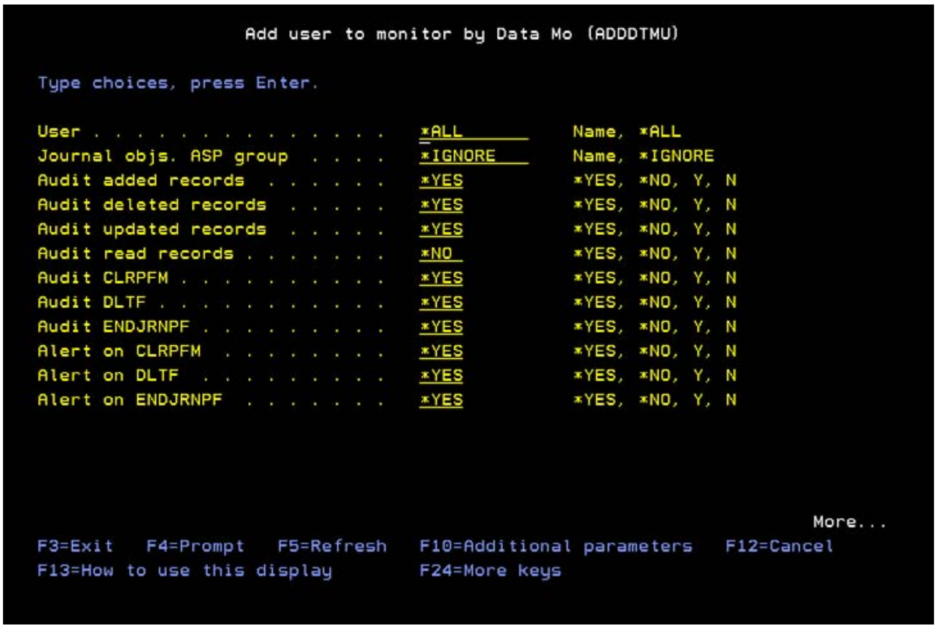
> "If you have selected to use the MINENTDTA parameter (i.e.*FILE) for the journal, then some of your journal entries entry-specific data will be minimized. The layout of the minimized record changes entries is completely different than the layout when the entry is not minimized. The data is not even recognizable nor is it readable as sophisticated hash techniques are used in addition to only operating on actual changed bytes. Therefore, if you want to use the journal as an audit mechanism, you may not want to choose this option (i.e. *FILE) for database physical files since you (i.e. Data Monitor) will not be able to read the actual change made."

See commands CRTJRN and CHGJRN for more details.

To add a new user to Data Monitor press F6 from the Work with monitored users screen. Another way to add a new file is to enter the following command:

T4DATAMON/ADDDTMU and press F4

The Add Data Monitor User command (ADDDTMU) adds a user configuration entry to the configuration list. The following screen appears:



```
                 Add user to monitor by Data Mo (ADDDTMU)

 Type choices, press Enter.

 User . . . . . . . . . . . . . .   *ALL           Name, *ALL
 Journal objs. ASP group  . . . .   *IGNORE        Name, *IGNORE
 Audit added records  . . . . . .   *YES           *YES, *NO, Y, N
 Audit deleted records  . . . . .   *YES           *YES, *NO, Y, N
 Audit updated records  . . . . .   *YES           *YES, *NO, Y, N
 Audit read records . . . . . . .   *NO            *YES, *NO, Y, N
 Audit CLRPFM . . . . . . . . . .   *YES           *YES, *NO, Y, N
 Audit DLTF . . . . . . . . . . .   *YES           *YES, *NO, Y, N
 Audit ENDJRNPF . . . . . . . . .   *YES           *YES, *NO, Y, N
 Alert on CLRPFM  . . . . . . . .   *YES           *YES, *NO, Y, N
 Alert on DLTF  . . . . . . . . .   *YES           *YES, *NO, Y, N
 Alert on ENDJRNPF  . . . . . . .   *YES           *YES, *NO, Y, N




                                                                  More...
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
```

Figure 41 – Add Data Monitor User screen I

Enter the user name of the user you want to monitor.

Next select the transactions you would like to monitor by setting each parameter to *YES or *NO. It is interesting to note that *Audit read records* is set to *NO by default. Remember that an individual user may read many files and transactions and auditing read records may have an impact on performance of your system, especially if you are monitoring *ALL users. Further note that you can select to monitor both *auditing values* and *alerts*.

Press PgDn and the second configuration screen for configuring additional parameters appears.

You can choose to change the additional parameters or simply use the default settings. They are not mandatory for Data Monitor to work.

By default new users appear in the Work with monitored users screen with Monitor Status *INACTIVE, Enabled *YES, and the values for Latest record date in historical are blank. When you start the monitor (F10) The Monitor Status will change to *ACTIVE.

Sometimes the Latest record date in historical field is filled immediately with an earlier date. This means that this user has been monitored in the past and for whatever reason the configuration entry was removed. However the monitoring data from the previous monitor was stored and now appeared when you started the monitor. Note that Latest record date in historical field is not provided for user `*ALL`.

Very Important
To start monitors successfully, make sure that the user profile of the Data Monitor jobs has access to the files added to Data Monitor. This user profile must also have `*ALL` authority for the journal objects and `*USE` authority for journal receiver objects.

The default user profile is `QPGMR`, but you can change it to add authorization for the required files. Use the `CHGJOBD` command in job descriptions `T4DATAMON/DTUMONITOR` and `T4DATAMON/STRDTUMON`.

4.3.1 Store Transaction If (F14)

Store Transaction If can be used to reduce the number of transactions stored in the Data Monitor databases. The expression used in Store Transaction if defines what transactions to store in the historical transaction log databases. For example, you could specify an ALEV expression to store only transactions for user NICOLE in Files A, B, and C. You can also use external business functions in this expression (*see section G.3 - External Business Functions on page 117*).

Store transaction if is available by pressing F14 in the Work with sensitive users screen.

Note
The store transaction if is the first expression for filtering data. If you are reusing formats, any expressions in these formats will be applied next. For a more detailed description, *see Appendix B: Journal Entry Process Flow on page 95* .

4.3.2 Monitoring Options (F15)

By default Data Monitor uses the **output library** `T4DATAMON` for format configuration files. However you may change the output library here. You can configure the library name where to store the physical files or databases of transactions filled by Data Monitor. Currently you can enter a specific library name or select `*PRDLIB`. Using this value the user indicates that the library name will be `T4DATAMON`.

You can also indicate whether you want to **reuse** the existing format for the files to be monitored or automatically generate a new one when Data Monitor finds the first journal entry for a specific file. By default Data Monitor reuses the existing file format. For details regarding file formats *see section 3.4 - Work With File Formats on page 17.*

Journal List allows you to configure what journals to monitor. Each entry is defined by a journal name and library. The default values are Journal `*ALL` and Library `*ALLUSR`, which will monitor all user journals, but automatically exclude system journals (journals starting with Q). For a description of the other variables press F1. The journal list applies to all sensitive users; it is not specific for a user.

Journals are processed in order of the journal list. To make sure that the process does not spend too much time processing a particularly large journal, use **Time per journal** to indicate how much time to spend on one journal before proceeding to process the next journal. By default time per journal is set to 30 seconds.

4.4 Starting/Restarting Monitors

You can use F10 – Start/Restart all monitors from the Work with monitored users screen to start or restart all monitors. Another way to start/restart all monitors is to enter the following command:

```
T4DATAMON/STRDTUMON
```

Only configuration entries that are enabled will be started/reinitialized when executing F10 or the command T4DATAMON/STRDTUMON.

Warning

If you need to journal a Data Monitor transaction file, such as CHLOC01P, we strongly recommend you use a journal that is not being monitored. Otherwise you will create an infinite loop.

Data Monitor saves transactions to these files. Because these files are being monitored, this transaction generates a new transaction to be monitored, which in turn generates a third transaction, and so on.

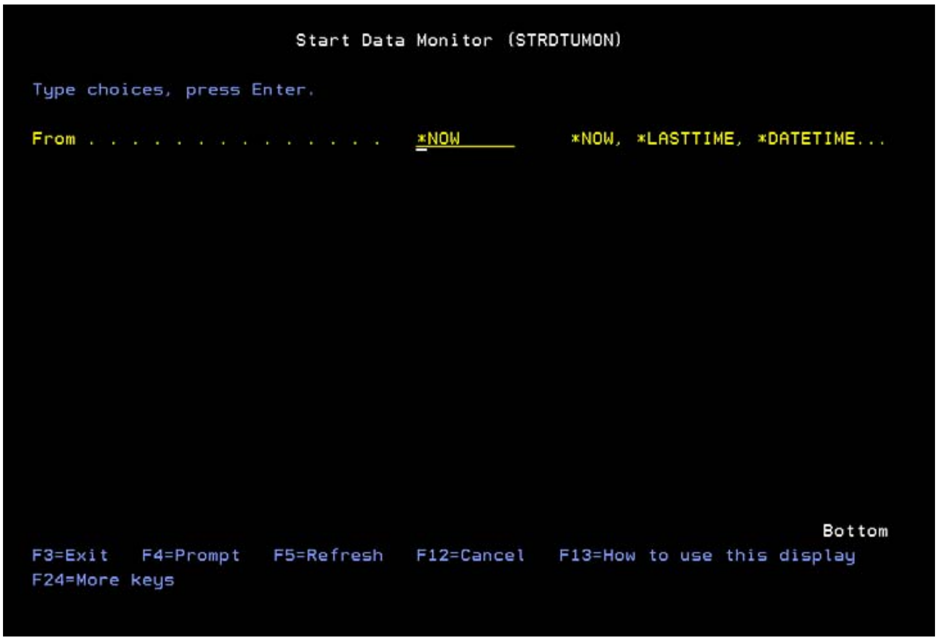
For example, an infinite loop will occur when:

- You configure *ALL for users *and* you journal the transaction files.
- DTUMONITOR job user is one of the monitored users and you journal the transaction files.

Note

This is not an issue for monitoring iSeries Files or Express.

When using F10 – Start/Restart all monitors a screen appears where you can indicate from what time you want to start the monitor. The options are *NOW, *LASTTIME, or from *DATETIME. The monitor status of the user will change, and an *ACTIVE status will appear if the process was successful.



```
                         Start Data Monitor (STRDTUMON)

 Type choices, press Enter.

 From . . . . . . . . . . . . .   *NOW          *NOW, *LASTTIME, *DATETIME...

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 42 – Enter the time you want to start the monitor

Default values are the values shown when using F10 – Start. They are also used for automatic starts after an IPL or after T4DATAMON subsystem start.

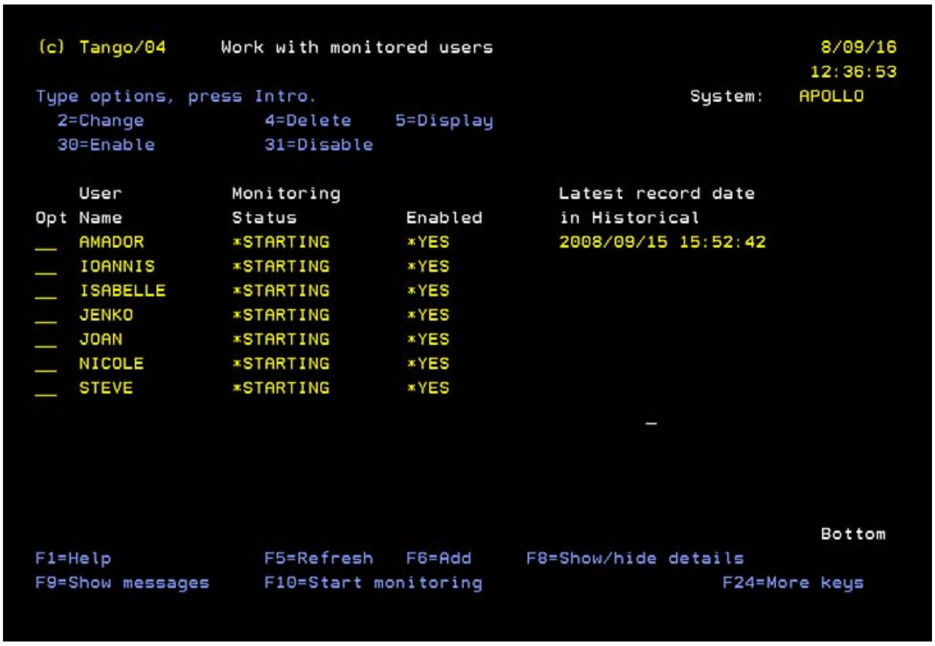
You can determine default values for from what point you want Data Monitor to start processing journal entries by using option 16 (Start settings for monitoring users) from the main Data Monitor menu (T4DATAMON/DTM_MENU). Another way to configure the starting point is to use the following command:

```
T4DATAMON/CHGDTUMON
```

Typically, the first time you run Data Monitor after installing the product, you will use the value *NOW to start processing journal entries from the current moment.

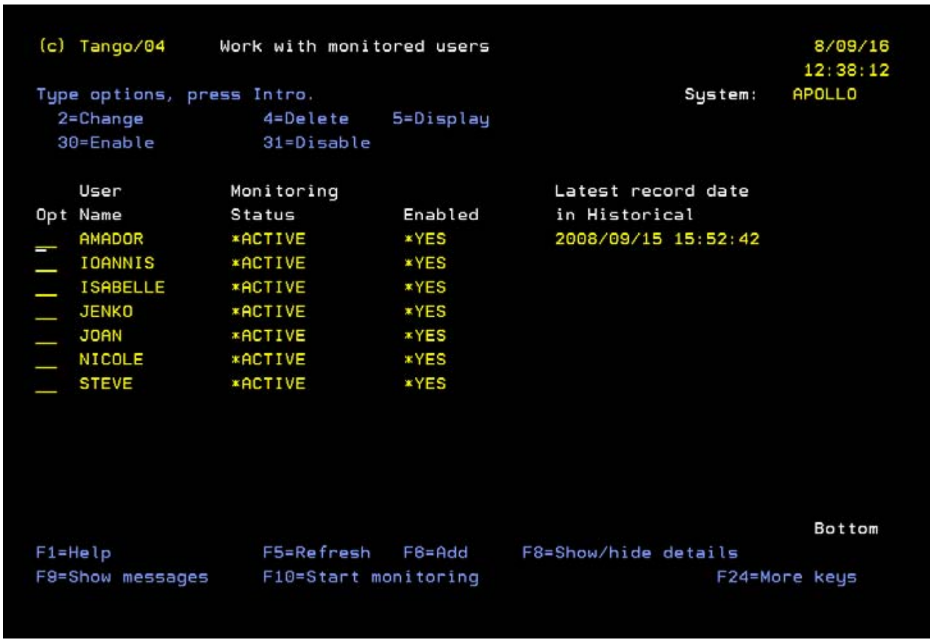
For normal operations once the monitor is up and running, you will set the default value to *LASTTIME. With this setting Data Monitor will start processing journal entries from the last processed entry in the journal. For example after an IPL, Data Monitor will start processing the first journal entry that has not yet been processed.

Only in special cases, for example when re-processing journal entries (*see Appendix C: How to Re-Process Entries on page 99*), will you set the value to *DATETIME to set the starting point to a specific date and time.



```
 (c) Tango/04     Work with monitored users                           8/09/16
                                                                     12:36:53
 Type options, press Intro.                              System:    APOLLO
   2=Change            4=Delete     5=Display
   30=Enable           31=Disable

     User          Monitoring                     Latest record date
 Opt Name          Status           Enabled       in Historical
 __  AMADOR        *STARTING        *YES          2008/09/15 15:52:42
 __  IOANNIS       *STARTING        *YES
 __  ISABELLE      *STARTING        *YES
 __  JENKO         *STARTING        *YES
 __  JOAN          *STARTING        *YES
 __  NICOLE        *STARTING        *YES
 __  STEVE         *STARTING        *YES



                                                                      Bottom
 F1=Help              F5=Refresh   F6=Add     F8=Show/hide details
 F9=Show messages     F10=Start monitoring                     F24=More keys
```

*Figure 43 – After pressing F10 monitor status for enabled entries changes to *STARTING.*



```
 (c) Tango/04     Work with monitored users                           8/09/16
                                                                     12:38:12
 Type options, press Intro.                              System:    APOLLO
   2=Change            4=Delete     5=Display
   30=Enable           31=Disable

     User          Monitoring                     Latest record date
 Opt Name          Status           Enabled       in Historical
 __  AMADOR        *ACTIVE          *YES          2008/09/15 15:52:42
 __  IOANNIS       *ACTIVE          *YES
 __  ISABELLE      *ACTIVE          *YES
 __  JENKO         *ACTIVE          *YES
 __  JOAN          *ACTIVE          *YES
 __  NICOLE        *ACTIVE          *YES
 __  STEVE         *ACTIVE          *YES



                                                                      Bottom
 F1=Help              F5=Refresh   F6=Add     F8=Show/hide details
 F9=Show messages     F10=Start monitoring                     F24=More keys
```

*Figure 44 – Monitors that were successfully started will show Monitor Status *ACTIVE.*

4.4.1 The T4DATAMON subsystem

All monitor jobs of the Data Monitor run under the `T4DATAMON` subsystem. Data monitor creates one job for each combination of active user and ASP group.

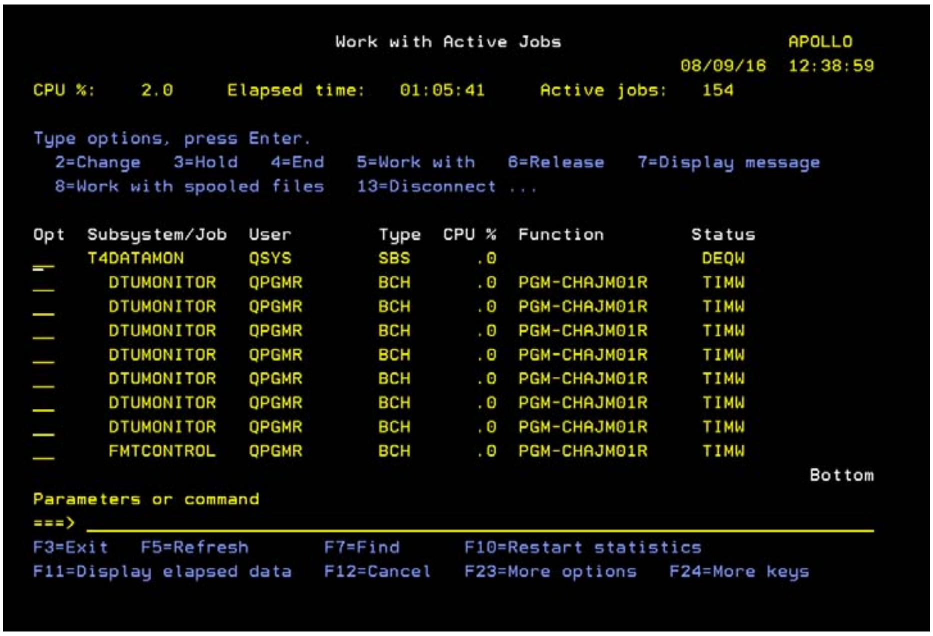
For example: the following jobs will be created for active sensitive users `JOAN` and `STEVE`:

JOAN	DTUMONITOR-JOAN
STEVE	DTUMONITOR-STEVE

Note that only one job is created for user `*ALL`, as follows:

***ALL**	DTUMONITOR-*ALL

Also you will find one job called `FMTCONTROL`. This is the job in charge of refreshing formats (For details regarding automatic refresh for formats, *see Chapter 5 - Data Monitor for iSeries Express on page 53*).

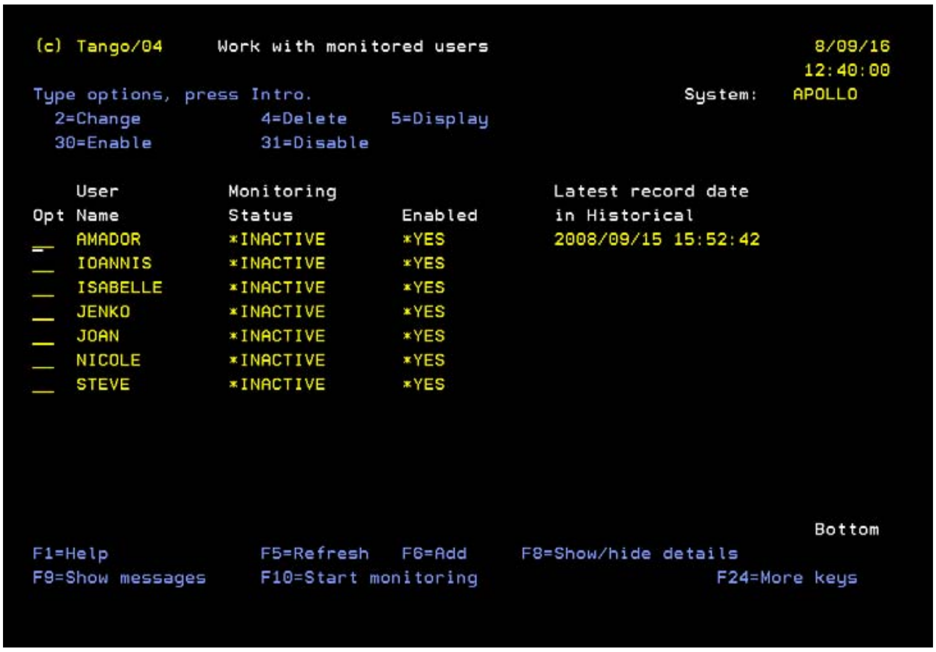


```
                          Work with Active Jobs                      APOLLO
                                                            08/09/16  12:38:59
CPU %:     2.0     Elapsed time:    01:05:41     Active jobs:    154

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
__   T4DATAMON      QSYS        SBS      .0                   DEQW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     DTUMONITOR   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
__     FMTCONTROL   QPGMR       BCH      .0  PGM-CHAJM01R     TIMW
                                                                     Bottom
Parameters or command
===> _________________________________________________________________________
F3=Exit   F5=Refresh        F7=Find      F10=Restart statistics
F11=Display elapsed data    F12=Cancel   F23=More options   F24=More keys
```

Figure 45 – Jobs for each sensitive user.

4.5 Ending Monitors

You can use F11 – End all monitors, from the Work with monitored user screen to end all monitors associated to monitored users. If the process was successful, the Monitor status will appear as `*INACTIVE`.



```
 (c) Tango/04     Work with monitored users                         8/09/16
                                                                   12:40:00
 Type options, press Intro.                               System:   APOLLO
   2=Change           4=Delete     5=Display
   30=Enable          31=Disable

     User          Monitoring                    Latest record date
 Opt Name          Status          Enabled       in Historical
 __  AMADOR        *INACTIVE       *YES          2008/09/15 15:52:42
 __  IOANNIS       *INACTIVE       *YES
 __  ISABELLE      *INACTIVE       *YES
 __  JENKO         *INACTIVE       *YES
 __  JOAN          *INACTIVE       *YES
 __  NICOLE        *INACTIVE       *YES
 __  STEVE         *INACTIVE       *YES

                                                                     Bottom
 F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
 F9=Show messages     F10=Start monitoring                   F24=More keys
```

*Figure 46 – Once monitors have been ended using F11 the Monitor Status changes to *INACTIVE.*

Chapter 5

Data Monitor for iSeries Express

Data Monitor allows you to pull once-off specific monitoring data on demand, for use in reports or the SmartConsole. Data Monitor for iSeries express can be used for various purposes, among which:

- Ad-hoc audit of a particular user's activities on the system, for example when you do not already have real-time monitoring of sensitive users set up in advance, or do not have the full version of Data Monitor for iSeries.
- Check the full auditing data. Frequently your real time monitoring is filtered for performance purposes. Data Monitor for iSeries Express allows you to check the full data for activity on a file or by a user right from the transaction log (data journal), for example in off-peak hours.
- Create ad-hoc reports containing data not included in your real-time monitors. As Data Monitor for iSeries Express allows you to use a separate data repository and data formats, you can use it to create specific reports containing different fields or filters next to your normal real-time audit reports.

Data Monitor for iSeries Express gathers data from journaled files for the specific period of time the user selects. This chapter describes how to use Data Monitor for iSeries Express.

5.1 Starting Data Monitor for iSeries Express

Data Monitor for iSeries is available from the main Data Monitor for iSeries menu (`DTM_MENU`).

To open Data Monitor for iSeries Express:

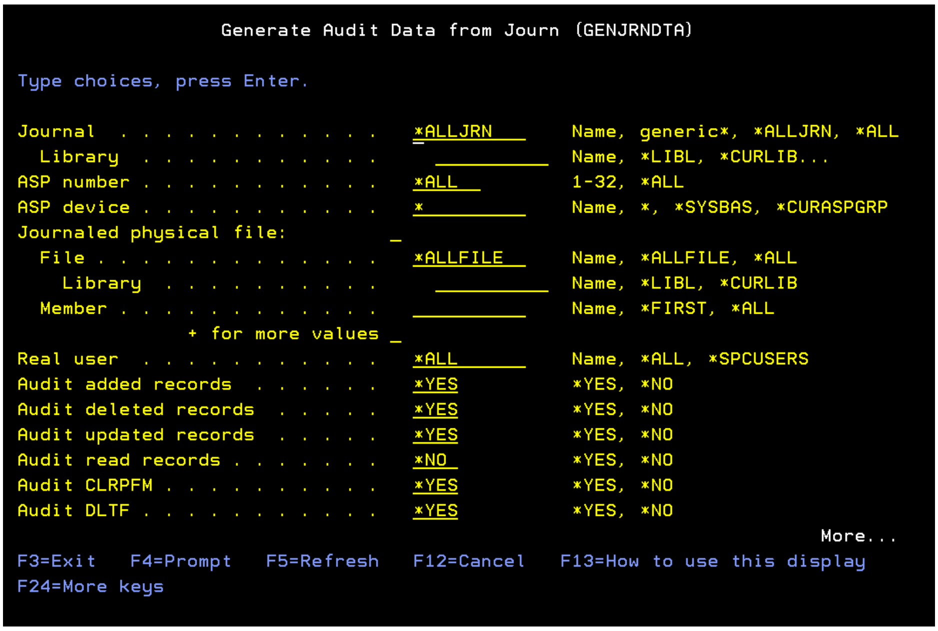
Step 1. Enter the following command:

```
Go T4DATATMON/DTM_CONFIG
```

Step 2. Select option **3** On Demand – Generate Audit Data from Journal (`GENJRNDTA`)

The Generate Audit Data from Journal (`GENJRNDTA`) screen opens (in fact it is a command called `GENJRNDTA` in library `T4DATAMON`).

Here you can configure a wide range of parameters, to define exactly what data to retrieve. Bear in mind that Data Monitor for iSeries Express works with all journaled transactions on your system, and depending on your auditing settings, may need to process large volume of data. We therefore strongly recommend that you carefully configure Data Monitor for iSeries Express to retrieve only essential data for your reporting purposes.



```
                   Generate Audit Data from Journ (GENJRNDTA)

 Type choices, press Enter.

 Journal  . . . . . . . . . . . .   *ALLJRN      Name, generic*, *ALLJRN, *ALL
   Library  . . . . . . . . . . .     __________ Name, *LIBL, *CURLIB...
 ASP number . . . . . . . . . . .   *ALL         1-32, *ALL
 ASP device . . . . . . . . . . .   *            Name, *, *SYSBAS, *CURASPGRP
 Journaled physical file:         _
   File . . . . . . . . . . . . .   *ALLFILE     Name, *ALLFILE, *ALL
     Library  . . . . . . . . . .     __________ Name, *LIBL, *CURLIB
   Member . . . . . . . . . . . .   __________   Name, *FIRST, *ALL
               + for more values _
 Real user  . . . . . . . . . . .   *ALL         Name, *ALL, *SPCUSERS
 Audit added records  . . . . . .   *YES         *YES, *NO
 Audit deleted records  . . . . .   *YES         *YES, *NO
 Audit updated records  . . . . .   *YES         *YES, *NO
 Audit read records . . . . . . .   *NO          *YES, *NO
 Audit CLRPFM . . . . . . . . . .   *YES         *YES, *NO
 Audit DLTF . . . . . . . . . . .   *YES         *YES, *NO
                                                                      More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 47 – Generate Audit Data from Journal (GENJRNDTA)

The first set of parameters helps identify exactly what files and users you are interested in.

Journal: we recommend you enter a specific journal name, however you can also use a generic name or the special values *ALL or *ALLJRN.

*ALLJRN refers to all journals on the system

*ALL is used in combination with Library values. When you enter *ALL, Data Monitor will select any journals that also have the specific value entered in the journal library parameter.

Library: *LIBL, *CURLIB, *USRLIBL, *ALLUSR, *ALL

ASP number: 1-32, *ALL

ASP Device: *CURASPGRP, *SYSBAS, *

Journaled physical file (File): enter the name or the special values *ALLFILE or *ALL.

*ALLFILE refers to all files on the system

*ALL works in combination with Journal Physical Library parameter value

Journaled physical file (Library): Name, *LIBL, *CURLIB

Journaled physical file (Member): Name, *FIRST, *ALL (+for more values)

Real User: enter a name or the special values *ALL or *SPCUSERS.

*ALL refers to all users

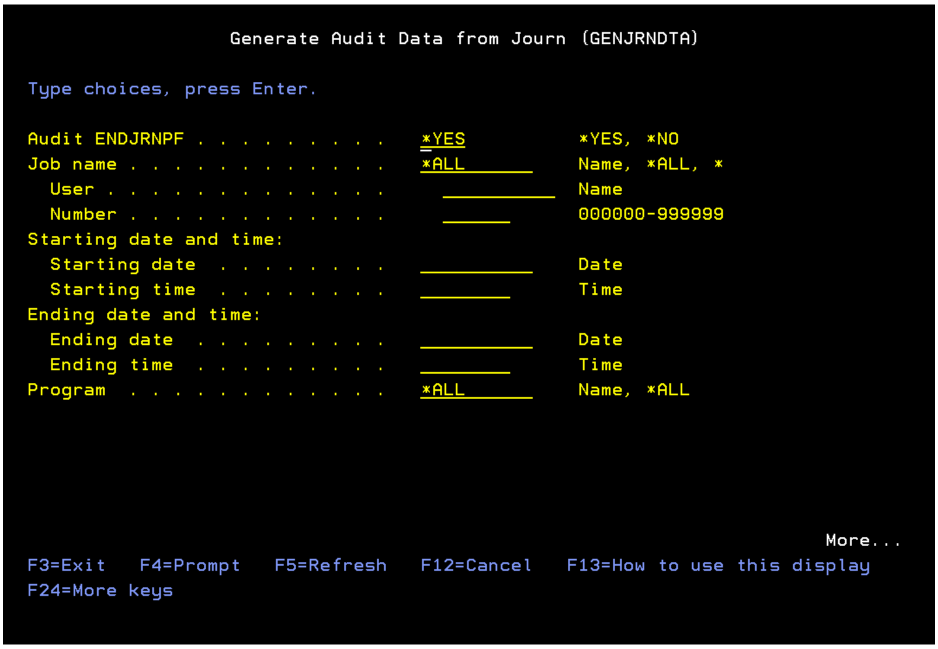
*SPCUSERS refers to the list of users configured in Data Monitor for iSeries Users. For example, if you have configured the users DAVID, IOANNIS, NICOLE, STEVE, … from the option *Work with monitored users* then you can use these users in Data Monitor for iSeries Express by selecting the value *SPCUSERS in the parameter USER.

Next indicate what transactions or commands you are interested in by setting the flag to YES or *NO.

Available parameters are:

- Audit added records
- Audit deleted records
- Audit updated records
- Audit read records
- Audit. CLRPFM
- Audit DLTF
- Audit ENDJRNPF

Press PgDn to see more values



```
                   Generate Audit Data from Journ (GENJRNDTA)

 Type choices, press Enter.

 Audit ENDJRNPF . . . . . . . . .   *YES          *YES, *NO
 Job name . . . . . . . . . . . .   *ALL          Name, *ALL, *
   User . . . . . . . . . . . . .                 Name
   Number . . . . . . . . . . . .                 000000-999999
 Starting date and time:
   Starting date  . . . . . . . .                 Date
   Starting time  . . . . . . . .                 Time
 Ending date and time:
   Ending date  . . . . . . . . .                 Date
   Ending time  . . . . . . . . .                 Time
 Program  . . . . . . . . . . . .   *ALL          Name, *ALL




                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 48 – Enter further details

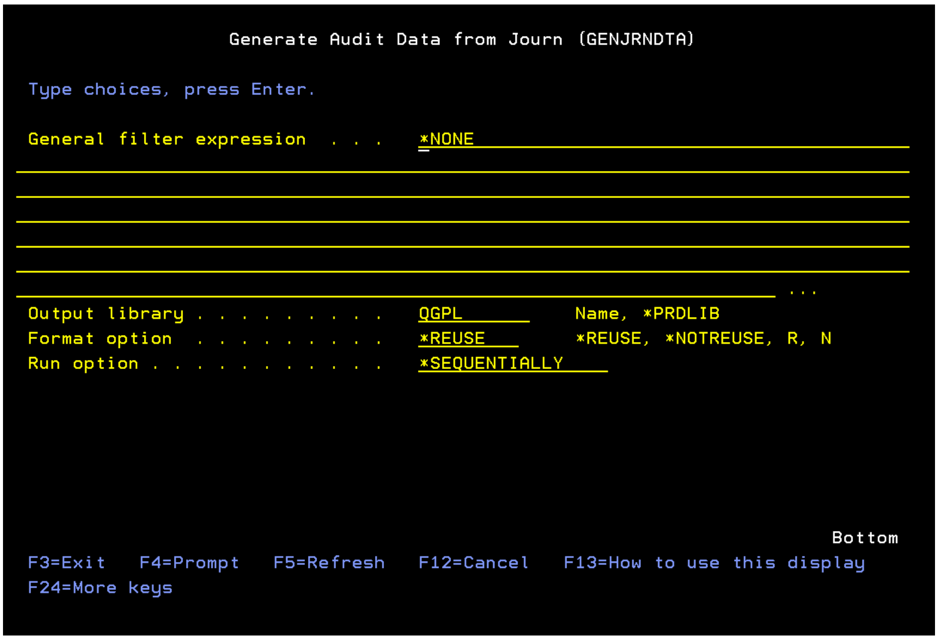
You can further narrow down your data collection by job. Here you can enter *job name, job user,* and *job number.*

Enter the *Start* and *End* of the period for which you want to retrieve data.

Note
The starting and ending date and time are mandatory fields.

The *program* field allows you to add a specific program name.

Press PgDn to see more values.



```
                 Generate Audit Data from Journ (GENJRNDTA)

 Type choices, press Enter.

 General filter expression . . .   *NONE
                                                                          ...
 Output library . . . . . . . . .   QGPL          Name, *PRDLIB
 Format option  . . . . . . . . .   *REUSE        *REUSE, *NOTREUSE, R, N
 Run option . . . . . . . . . . .   *SEQUENTIALLY

                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 49 – General filter expression

General Filter Expression: offers a first level ALEV filter. You can also use external business functions in your expression. For details *see section G.3 - External Business Functions on page 117*.

To minimize the load on the system it is important to define this configuration carefully, as you may need to process large volumes of data, particularly when reads are involved. The better you define your configuration the better use you will make of your resources.

Note
The general filter expression is the first expression for filtering data. If you are reusing formats (*REUSE), any expressions in these formats will be applied next. For a more detailed description, *see Appendix B: Journal Entry Process Flow on page 95.*

Finally you can indicate where and how to store the collected data

First enter the desired *output library* name or *PRDLIB in the Output Library field.

Next indicate whether you want to reuse the existing file *format* or want to create one based on the database file.

The *run option* (RUNOPTION): currently *SEQUENTIALLY is the only valid value.

5.2 Limitations

Data Monitor for iSeries Express can only monitor data available on the system in active journal receivers, whereas Data Monitor for iSeries – Base allows you to store data in a separate repository for use in the future.

If you find your journals grow quickly and you frequently need to delete data form jour journal receivers it may be more practical to use the full Data Monitor for iSeries – Base. Then you do not need to maintain all the data in your journal receivers, but can transfer data to a separate data repository.

Advantages of Data Monitor for iSeries Base include:

- Real-time alerts on any suspicious activity
- Off-line data storage of auditing data

- The ability to monitor business activity, by extracting key performance indicators in real-time, for purposes of business intelligence, in addition to security purposes.

Upgrading to Data Monitor for iSeries - Base is easy:

- Simply contact your Tango/04 Business Representative to get the required product activation key, and enter it in your existing installation of Data Monitor for iSeries.

Note
Existing users of Data Monitor for iSeries Express are entitled to a special upgrade price.

Chapter 6
Handling formats in Data Monitor for iSeries

For Data Monitor to use raw (unformatted) data of transactions stored in IBM journals, it must translate the data to a valid layout for its historical transaction log. Data Monitor uses formats (or schemas) to convert the raw data.

A format (or schema) includes a list of field definitions for the sensitive file and a list of key fields for ordering the records in transaction log reports.

The way Data Monitor reads these *formats* differs slightly depending on whether you monitor sensitive files, sensitive users, or use Data Monitor for iSeries Express. Each method is described in detail in the following sections.

6.1 Data Monitor for Sensitive Files

When you manually select the sensitive files to monitor, a format is retrieved and stored for every file. You can also create multiple formats for a sensitive file. For details, *see section 3.4.1 - Adding a Format to a File on page 19* and *see section 3.4.2 - Changing Format Definitions on page 20*.

When Data Monitor detects a sensitive file may have changed (for example a field has been added or deleted) it uses the *Automatic refresh for formats* function to load a new format with the changes. For details, see Automatic Refresh for Formats below.

6.2 Data Monitor for Sensitive Users

When monitoring sensitive users Data Monitor will read transactions from journals of any file in your system. Here it is impractical to manually load formats for all files in your system before starting monitoring.

Therefore, Data Monitor uses the *Load Formats on the fly* function:

- When a monitor finds the first journal entry for a specific file, it loads the format from this file and uses it to store this transaction and future ones.

In the case of sensitive files where you have added formats manually, you can reuse the existing formats when Data Monitor for sensitive users finds transactions of these files. Configuring formats for sensitive files with *Work with file formats* obviously gives you more control over the format than in the formats created by the *Load Formats on the fly* function.

As when monitoring sensitive files, if any of the monitors detects that a sensitive file may have changed (for example a field has been added or deleted) Data Monitor uses the *Automatic refresh for formats* function to load a new format including the changes. For details *see section 6.4 - Automatic Refresh for Formats below*.

6.3 Data Monitor for iSeries Express

Data Monitor Express works exactly as Data Monitor for sensitive users regarding formats. It uses *Load Formats on the fly* function for loading formats, can reuse existing formats of sensitive files, and uses *Automatic refresh for formats* function to load a new format including changes.

6.4 Automatic Refresh for Formats

Data Monitor detects a change when an end journal physical file and start journal physical file events are performed on a file. When the file already has one or more formats, the automatic refresh for format feature creates a new format including the changes. This new format is loaded just as in the *Load Formats on the fly* function and then updated with the configuration of the nearest existing format. In short you can understand the whole process as a refresh of the old format.

Example of the automatic refresh for formats

The format for file `LIB/FILE` contains 2 fields: `FIELDA` and `FIELDB`.

Data Monitor detects this file has changed: a new field, `FIELDC`, has been added.

Data Monitor creates and updates the new format:

- The new format will retrieve values for Expression to use format, Expression to store, and Preprocess expression from the old format (for details of each expression, *see section 3.4.3 - Using expressions in formats on page 21*).
- For `FIELDA` and `FIELDB`, the new format will retrieve details of values Print As, Audit transaction if, Save field if, and Protection field from the old format (For details of these parameters, *see section 3.5 - Work With File Format Details on page 22* and *section 3.5.3 - View 3 - Auditing and Protection Details on page 27*).
- `FIELDC` is a new field and will maintain all its values from the new format.

Note
As `*VIRTUAL` fields are copied from the old format to new format with no modifications; make sure that these fields are still valid. If not, you will need to change them manually.

6.4.1 Key fields and Refresh for formats

If the old format was created with parameter *initial key list* (`ADDDTMF` command) or *key list* (`ADDDTMFMT` or `CHGDTMFMT` commands) equal to the monitored file (or `*MONFILE`), then key fields in the old format will be kept in the new format. So any manual change you made regarding key fields in old format will be maintained.

If the old format was created with parameter *initial key list* (`ADDDTMF` command) or *key list* (`ADDDTMFMT` or `CHGDTMFMT` commands) different from the monitored file (or `*MONFILE`), then automatic refresh for formats will read the list of key fields from the *key list file* for the new format. So any manual changes you made regarding key fields in the old format would be lost.

When automatic refresh does not work

If, anytime, automatic refresh for formats does not work as you would desire with key fields, you can use `CHGDTMFMT` (Change Format Definition) to prepare the key list parameter and then follow the following steps to repeat/correct the automatic refresh for formats:

- Write down the *Format from* date of the formats you consider invalid and then delete them. For further details *see section 3.4 - Work With File Formats on page 17*.
- Make any necessary changes to formats that have to be automatically refreshed. See above for more info.

Reprocess entries from the date *Format from* you have written down in the earlier step. For further details *see Appendix C: How to Re-Process Entries on page 99*.

© 2013 Tango/04 Computing Group

Chapter 7
Cleaning Data Monitor files

Data Monitor needs to work with different files that can store a huge amount of data, processed from journal receivers. It is very important to keep disk space under control, by regularly cleaning the Data Monitor historical transaction log. We recommend you only remove unused data or data that has been stored elsewhere.

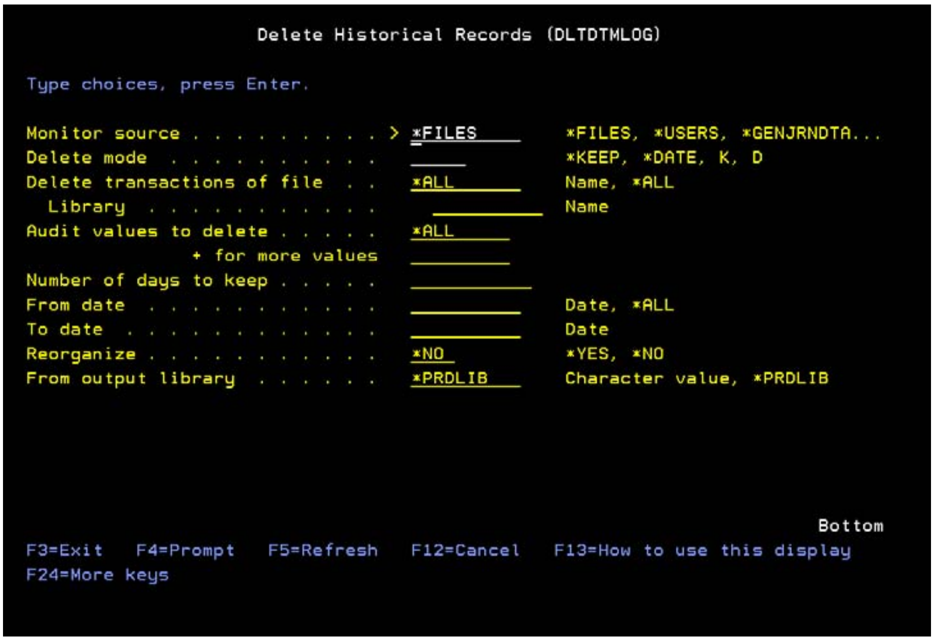
Use the following command to clean Data Monitor Files:

```
T4DATAMON/DLTDTMLOG
```

7.1 Delete Mode

In delete mode you can filter which records to delete from the historical transaction database based on the monitor type which is the source of these records. The user can indicate one of the following special values:

- `*FILES`: Transactions will be deleted for any monitor based on files configured from Work with monitored files option.
- `*USERS`: Transactions will be deleted for any monitor based on users configured from Work with monitored users option
- `*GENJRNDTA`: Any transactions generated with command Generate Audit Data from Journal (`GENJRNDTA`) will be deleted.
- `*ALL`: Transactions will be deleted for all types of monitors.



```
                 Delete Historical Records (DLTDTMLOG)

 Type choices, press Enter.

 Monitor source . . . . . . . . . > *FILES____    *FILES, *USERS, *GENJRNDTA...
 Delete mode  . . . . . . . . . .   _____         *KEEP, *DATE, K, D
 Delete transactions of file  . .   *ALL______    Name, *ALL
   Library  . . . . . . . . . . .     __________  Name
 Audit values to delete . . . . .   *ALL_____
               + for more values    _________
 Number of days to keep . . . . .   ___________
 From date  . . . . . . . . . . .   __________    Date, *ALL
 To date  . . . . . . . . . . . .   __________    Date
 Reorganize . . . . . . . . . . .   *NO_          *YES, *NO
 From output library  . . . . . .   *PRDLIB___    Character value, *PRDLIB




                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

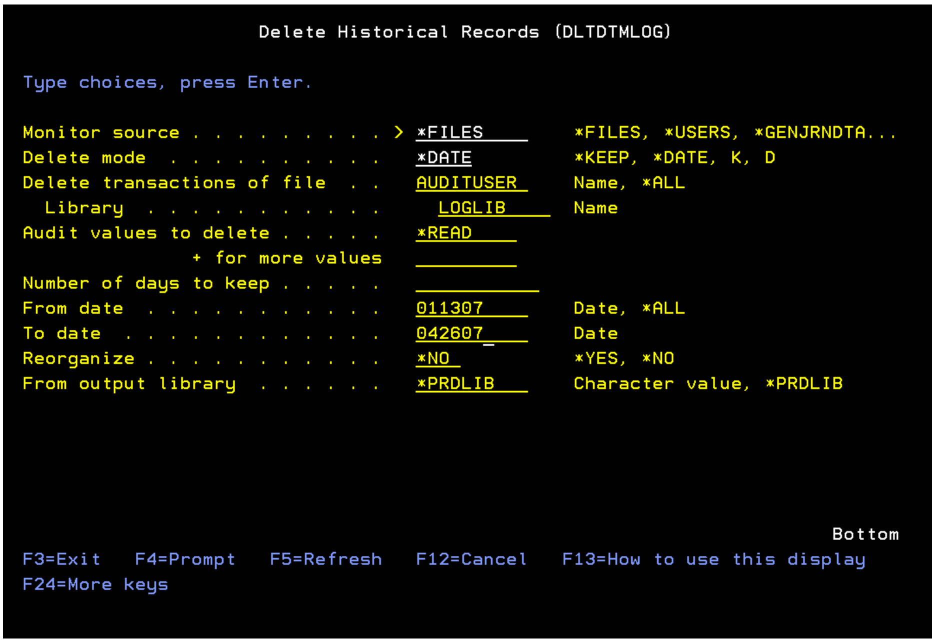
Figure 50 – Delete Historical Records (DLTDTMLOG)

As in all Tango/04 cleaning commands, there are two delete modes:

- Keep mode (`*KEEP` - K)
- Date mode (`*DATE` - D)

In the Keep mode you can specify the number of days you want to keep records, the audit values to delete, and the transactions file. When using Keep mode only the records with a date older than the current date minus (-) K days are deleted. The result is that only records for the last K days are left in the database.

In Date mode you can specify the date range for which you want to delete records, the audit values to delete and the transactions file. When using Date mode all records that fall in the date range are deleted *see section 7.2 - Delete Transaction of File below*.



```
                 Delete Historical Records (DLTDTMLOG)

 Type choices, press Enter.

 Monitor source . . . . . . . . . > *FILES____    *FILES, *USERS, *GENJRNDTA...
 Delete mode  . . . . . . . . . .   *DATE         *KEEP, *DATE, K, D
 Delete transactions of file  . .   AUDITUSER_    Name, *ALL
   Library  . . . . . . . . . . .     LOGLIB____  Name
 Audit values to delete . . . . .   *READ____
               + for more values    _________
 Number of days to keep . . . . .   ___________
 From date  . . . . . . . . . . .   011307____    Date, *ALL
 To date  . . . . . . . . . . . .   042607____    Date
 Reorganize . . . . . . . . . . .   *NO_          *YES, *NO
 From output library  . . . . . .   *PRDLIB___    Character value, *PRDLIB




                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 51 – Delete Historical Records (DLTDTMLOG), Date mode

7.2 Delete Transaction of File

By default records are deleted for all files according to the delete mode as described above. However it is also possible to delete records for a particular file only. You can specify the file and library for which you want to delete historical records.

7.3 Selecting Audit Values to Delete

In addition the user can indicate the category of records to delete from the historical transactions database. The type of operation that the record represents in the database defines this category. Valid audit values are:

*ADD	*UPDATE	*DLTF	*READ
*DELETE	*CLRPFM	*ENDJRNPF	

7.4 Reorganizing Files

You can also reorganize files to free up disk space left by the deleted records. Note that when you set Reorganize to *YES the operation will be slower and any jobs locking the database file will be ended immediately, including all Data Monitor jobs.

7.5 Output Library

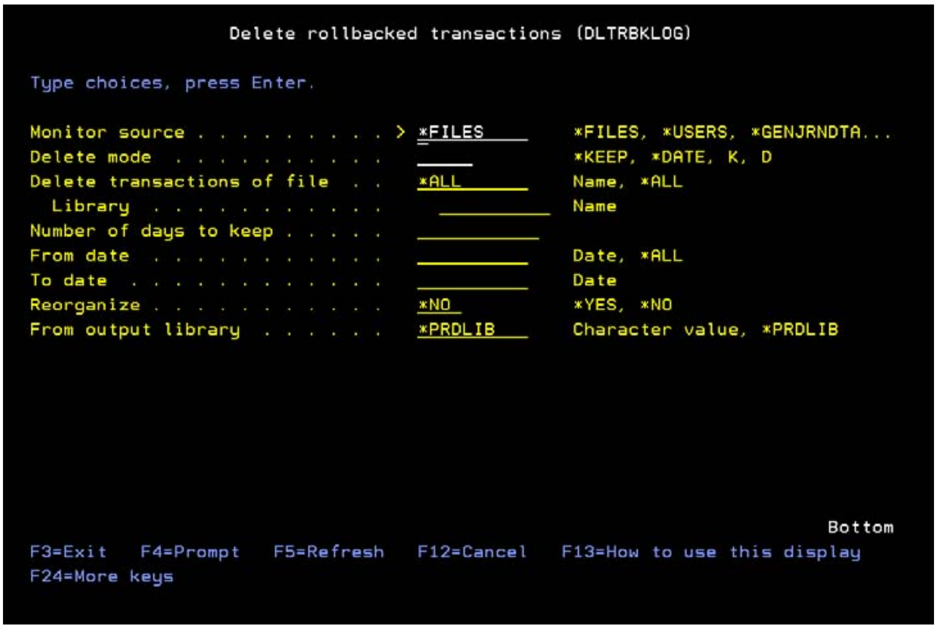
You can specify the library where the output file that contains transactions to be deleted is located. You can specify a specific name or use the special values *PRDLIB. When you enter *PRDLIB transactions will be deleted for the output file located in the default product library, T4DATAMON.

7.6 Delete rolled back transactions

As a transaction that has been rolled back does not represent a real change in the database for auditing purposes it is useful to remove these transactions from the database.

You can delete rolled back transactions using option **24** – delete rolled back transactions of historical records from the main Data Monitor Menu (DTM_Menu).

To delete a rolled back transaction indicate the keep mode (*see also section 7.1 - Delete Mode on page 61*), the file from which to delete the rolled back transactions and the library of the file.



```
                    Delete rollbacked transactions (DLTRBKLOG)

 Type choices, press Enter.

 Monitor source . . . . . . . . . > *FILES       *FILES, *USERS, *GENJRNDTA...
 Delete mode  . . . . . . . . . .   _____        *KEEP, *DATE, K, D
 Delete transactions of file  . .   *ALL         Name, *ALL
   Library  . . . . . . . . . . .     __________ Name
 Number of days to keep . . . . .   ___________
 From date  . . . . . . . . . . .   __________   Date, *ALL
 To date  . . . . . . . . . . . .   __________   Date
 Reorganize . . . . . . . . . . .   *NO          *YES, *NO
 From output library  . . . . . .   *PRDLIB      Character value, *PRDLIB

                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 52 – Delete rolled back transactions

Chapter 8

360º data enrichment for 360º Auditing and Control

This unique new feature gives unprecedented flexibility to create virtual fields and enrich audited transactions with data from secondary tables. New, easy to use, fast functions provide for convenient data retrieval from local or remote databases.

The new functionality simplifies enormously the burden of auditing, since it makes data much more readable. Moreover, suspicious transactions are easier to spot, while comprehensive real time controls can be set on business data to greatly reduce corporate risk.

The benefits of the enrichment are:

- **Readable audit reports**: instead of dealing with arcane codes and product or account numbers, reports can show real customer names, product names, etc.
- **Contextual business data**: by enriching the original transaction data, additional information about the transaction can be produced, such as maximum discount ranges, customer type, territory, address, etc.
- **Employee identification**: additional data about the employee executing the transaction can be retrieved from a master table. For instance, user `JSMITH` can be correctly reported as `JOHN SMITH` from `ACCOUNTING`. More information such as physical location, social security number, or whether the employee is on vacation or not can be retrieved from secondary tables, and alerts can be produced accordingly.
- **Comprehensive Real time alerts**: any type of control can be done in real time and sent to a message queue (or the VIRTUAL Message Center SmartConsole) regarding the transaction or its enriched data. For instance, an alert can be sent whenever a customer of certain type receives a discount that is higher than the maximum discount percentage allowed for the type of customer, transaction, product, etc., even if this data is in secondary tables. If needed, complex scripting logic and correlation can be used, and even custom RPG/COBOL program logic can be reused if required.
- **KPI (Key Performance Indicator) monitoring**: numerical data (such as "sales per day") can be easily monitored as the changes are produced instead of continuously running heavy SQL statements, reducing both the resource consumption and the time to react to business events. KPIs can be exported to the SmartConsole and the DashboardServer (this requires the Real Time version of Data Monitor for iSeries, which will be available in future releases of Data Monitor for iSeries).

- **Ease of maintenance**: the new functions are easy to learn and maintain, no error-prone database triggers are necessary, and since no data needs to be hard-coded, all the information is automatically updated from the live data.
- **Maximum control**: as all kinds of business and security rules can be set with low resource consumption, maximum control can be achieved easily.
- **Easier Compliance with Regulations and Best Practices**: enhanced control simplifies the burden of complying with regulations such as Sarbanes Oxley, PCI, etc. or best practice frameworks such as COBIT.
- **Reduced time required to audit and control**: as practically any kind of control can be automated, and reports are easier to read, the required time to audit and control the whole corporate data is reduced enormously.

Simple example of 360º data enrichment

A transaction may include a field containing Product Code values; however the user does not know which product the product code refers to, as this information is only available in a secondary table.

An ALEV function can be used to automatically link the information from the table containing the Product Code details and include this information directly in the report, thus the user will not need to manually look up this information after producing a report of the transaction.

8.1 ALEV functions used for 360º data enrichment

Enrichment allows you to retrieve any value from any field of any table in your system, including remote systems. This is achieved using ALEV functions that can query fields from other tables available in the system, and return them as ALEV Variables. For more information regarding ALEV variables and functions, *see Appendix G: ALEV Variables and Functions for Expressions on page 109*.

8.1.1 ALEV functions for local databases

The following ALEV functions can be used to retrieve data from local databases:

- `FUNCTION RunSQL(SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)`
- This function runs the SQL statement provided by `SQLSTMT`, returns the `SQLCODE` value as parameter and it if is a `SELECT SQL` it will work in the same way as the following function `GetRow`. This means that it also returns an ALEV array with values of fields requested between `SELECT` and `FROM` clause. This array is indexed by field name and also by a numerical index.
- `FUNCTION GetRow(SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)`
- This function runs the SQL statement provided by `SQLSTMT` only if it is a `SELECT` statement (in any other case it returns an error). It returns the `SQLCODE` value as parameter and it also returns an ALEV array with values of fields requested between `SELECT` and `FROM` clause. This array is indexed by field name and also by a numerical index.
- `FUNCTION GetCol(LIBRARY STRING, FILE STRING, FIELD STRING, Var SQLCODE INTEGER, [KEYFIELD1 STRING], [KEYVALUE1 Variant], [KEYFIELD2 STRING], [KEYVALUE2 Variant], …) : Variant`
- This function retrieves the value of the field `FIELD`, in the first row of the file `LIBRARY/FILE`, which accomplishes the condition: `KEYFIELD1 = KEYVALUE1 and KEYFIELD2 = KEYVALUE2` and... . Therefore, it differs from the two previous functions by providing a kind of native interface rather than a SQL interface.

Note

The GetCol function can only retrieve one field value, while the GetRow function is capable of retrieving one or more field values.

8.1.2 ALEV functions for remote databases

The following ALEV functions can be used to retrieve data from remote databases:

- `FUNCTION RunRMTSQL(RDBNAME STRING, SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)`
- This works in the same way as the RunSQL function but in RDBNAME you should provide a remote relational database name (use system command WRKRDBDIRE).
- `FUNCTION GetRMTRow(RDBNAME STRING, SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)`
- This works in the same way as the GetRow function but in RDBNAME you should provide a remote relational database name.
- `FUNCTION GetRMTCol(RDBNAME STRING, LIBRARY STRING, FILE STRING, FIELD STRING, Var SQLCODE INTEGER, [KEYFIELD1 STRING] [KEYVALUE1 Variant],[KEYFIELD2 STRING], [KEYVALUE2 Variant], …) : Variant`
- This works in the same way as the GetCol function but in RDBNAME you should provide a remote relational database name.

Each of the functions for retrieving information from remote databases use a CONNECT SQL statement, so as long as you do not need a username and password in the other SQL interface (STRSQL for example), you will be able to connect with this ALEV function.

Note

To connect to a remote database, the Data Monitor job user must have sufficient permissions to CONNECT TO RDATABASE.

The default user profile is QPGMR, but you can change it to add authorization for the required files. Use the CHGJOBD command to change Data Monitor job descriptions.

Remote databases requiring passwords cannot be accessed for security reasons.

The sqlcode variable

You can use the sqlcode variable to control your scripts execution when encountering errors.

When accessing databases, there is a wide-range of possibilities which can lead to access problems, particularly when working with remote machines. For example, you could be trying to access a database in an offline system; or the format of a database could have changed; or a specific value that you expected to be in a specific record in a database could have been unexpectedly changed.

Using sqlcode variables you can now include the details of these errors when retrieving information, in your scripts, whatever the reason for the error may be.

8.2 How to use the ALEV functions for enrichment

In this section we will explain how to use ALEV functions in Data Monitor for iSeries to enrich audited transactions with data from secondary tables.

Step 1. Enter the command:

```
GO T4DATAMON/DTM_MENU
```

Step 2. Enter option **1**, to open Data Monitor Settings.

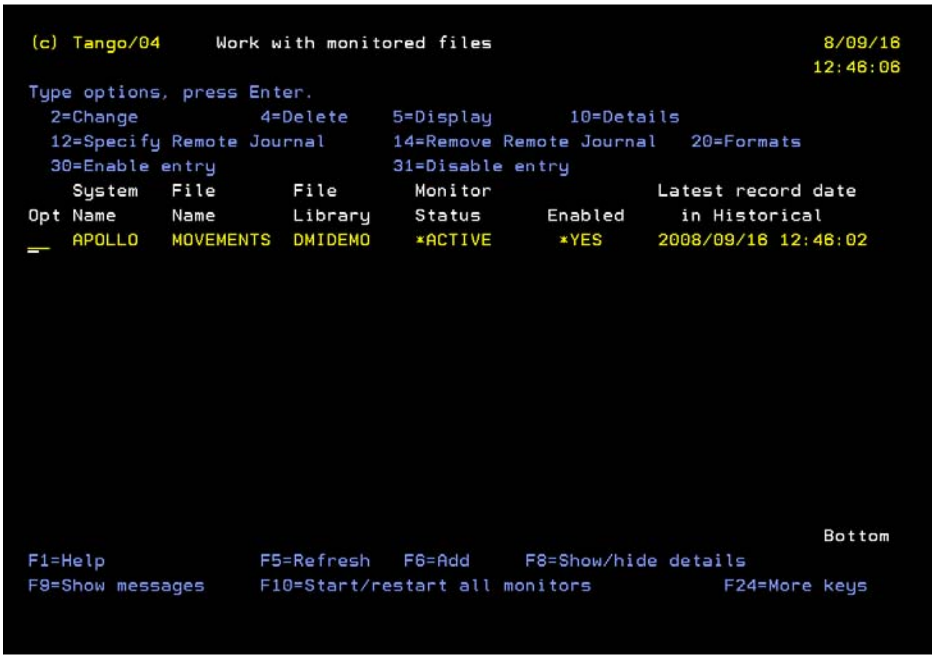
Step 3. Enter option **1** again, to work with monitored files. Here you can add a monitored file.

We are going to write a script using a preprocess expression in this instance.

Note

An ALEV function (or functions) is used in ALEV expressions in Data Monitor for iSeries. You can use expressions such as: Use format If; Store Transaction If; Preprocess; Audit Transaction if; Save Field If; and more.

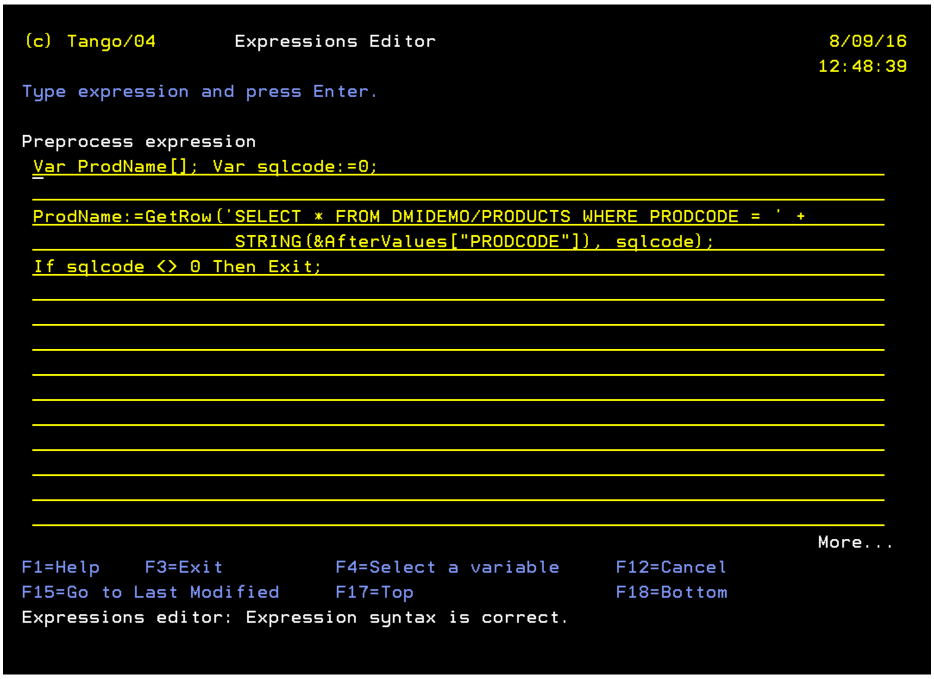
Step 4. Enter option **20**, to work with formats.



```
(c) Tango/04     Work with monitored files                                 8/09/16
                                                                          12:46:06
Type options, press Enter.
  2=Change           4=Delete     5=Display       10=Details
  12=Specify Remote Journal       14=Remove Remote Journal    20=Formats
  30=Enable entry                 31=Disable entry
    System    File        File        Monitor                Latest record date
Opt Name      Name        Library     Status      Enabled      in Historical
__  APOLLO    MOVEMENTS   DMIDEMO     *ACTIVE      *YES      2008/09/16 12:46:02









                                                                            Bottom
F1=Help              F5=Refresh   F6=Add      F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors             F24=More keys
```

Figure 53 – Enter option 20 to work with formats

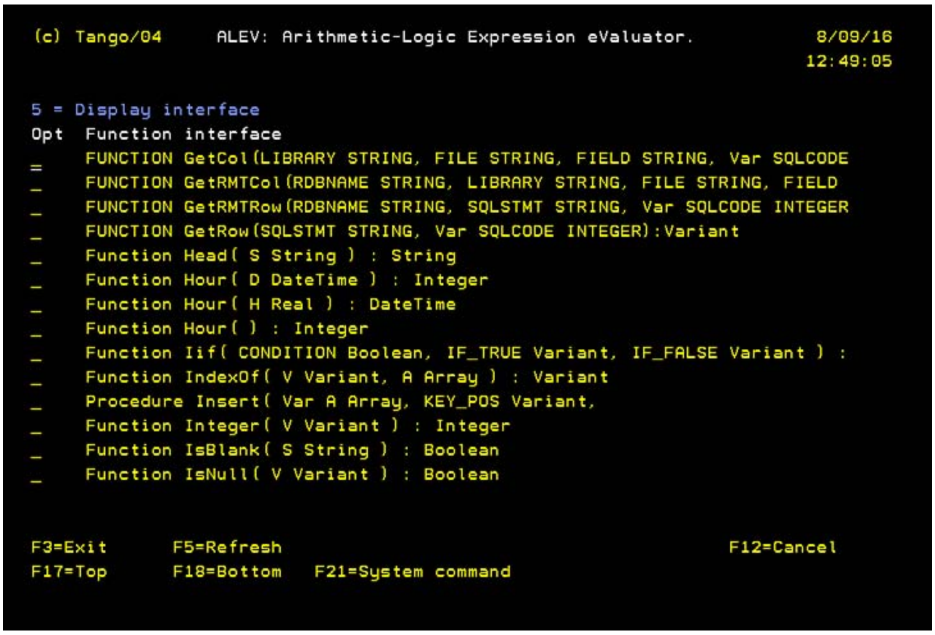
Step 5. Enter option **40**, to work with Preprocess expressions in the Expressions Editor.



```
(c) Tango/04        Expressions Editor                          8/09/16
                                                               12:48:39
Type expression and press Enter.

Preprocess expression
 Var ProdName[]; Var sqlcode:=0;

 ProdName:=GetRow('SELECT * FROM DMIDEMO/PRODUCTS WHERE PRODCODE = ' +
                  STRING(&AfterValues["PRODCODE"]), sqlcode);
 If sqlcode <> 0 Then Exit;











                                                              More...
F1=Help     F3=Exit          F4=Select a variable     F12=Cancel
F15=Go to Last Modified      F17=Top                  F18=Bottom
Expressions editor: Expression syntax is correct.
```

Figure 54 – Working with the Preprocess Expressions. The above script is similar to one that would be used in the example on page 74

Step 6. Press F1 then press ENTER to see a list of all the ALEV functions available. You should see the new ALEV functions.



```
 (c) Tango/04     ALEV: Arithmetic-Logic Expression eValuator.          8/09/16
                                                                       12:49:05

 5 = Display interface
 Opt  Function interface
  =    FUNCTION GetCol(LIBRARY STRING, FILE STRING, FIELD STRING, Var SQLCODE
  _    FUNCTION GetRMTCol(RDBNAME STRING, LIBRARY STRING, FILE STRING, FIELD
  _    FUNCTION GetRMTRow(RDBNAME STRING, SQLSTMT STRING, Var SQLCODE INTEGER
  _    FUNCTION GetRow(SQLSTMT STRING, Var SQLCODE INTEGER):Variant
  _    Function Head( S String ) : String
  _    Function Hour( D DateTime ) : Integer
  _    Function Hour( H Real ) : DateTime
  _    Function Hour( ) : Integer
  _    Function Iif( CONDITION Boolean, IF_TRUE Variant, IF_FALSE Variant ) :
  _    Function IndexOf( V Variant, A Array ) : Variant
  _    Procedure Insert( Var A Array, KEY_POS Variant,
  _    Function Integer( V Variant ) : Integer
  _    Function IsBlank( S String ) : Boolean
  _    Function IsNull( V Variant ) : Boolean


 F3=Exit      F5=Refresh                                      F12=Cancel
 F17=Top      F18=Bottom    F21=System command
```

Figure 55 – List of ALEV functions, including some of the new functions for enrichment

Step 7. In the Expressions Editor write an expression using the new ALEV functions. Exit the editor and enter Y to save the expressions when prompted.

Step 8. You must restart the monitor to be able to use the new expressions. The monitor will then start to work with the monitored file and execute the preprocess expressions.

You can check if something is not working properly by checking error messages in the product message.

> **Warning**
> If you are working with databases monitored by Data Monitor for iSeries, you have to be careful not to create infinite loops of events to be monitored. For further details, *see Appendix K: Frequently Asked Questions on page 147*.

8.3 Create a virtual field for enrichment

We can create virtual fields that do not already exist in the databases that we are working with. These are extremely useful as they can be used to provide additional information that creates more enriched and informative reports.

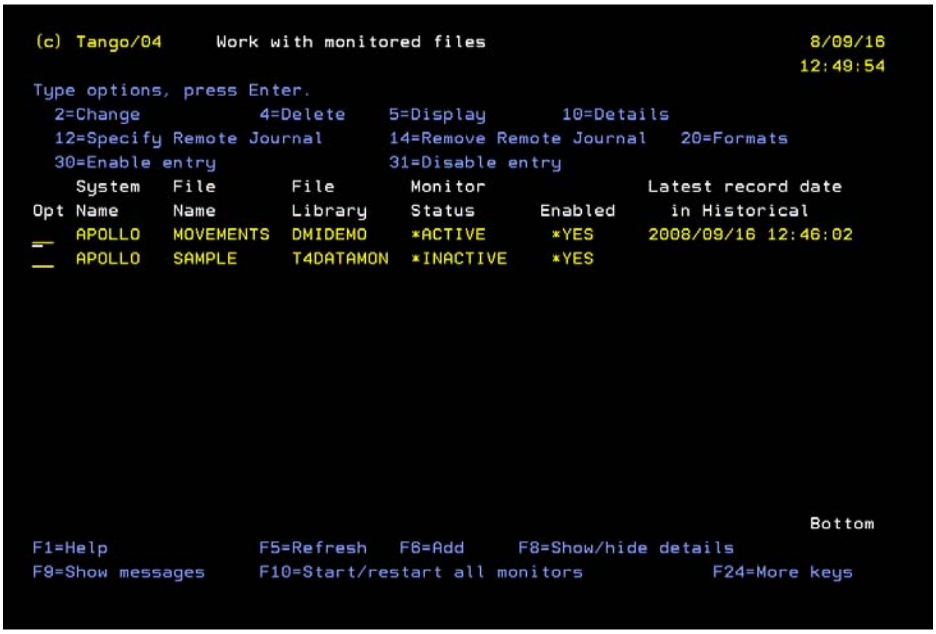
To create a virtual field for enrichment:

Step 1. Enter the command:

```
GO T4DATAMON/DTM_CONFIG
```

Step 2. Enter option **1**, to work with monitored files.

Step 3. Enter option **20**, to work with formats.

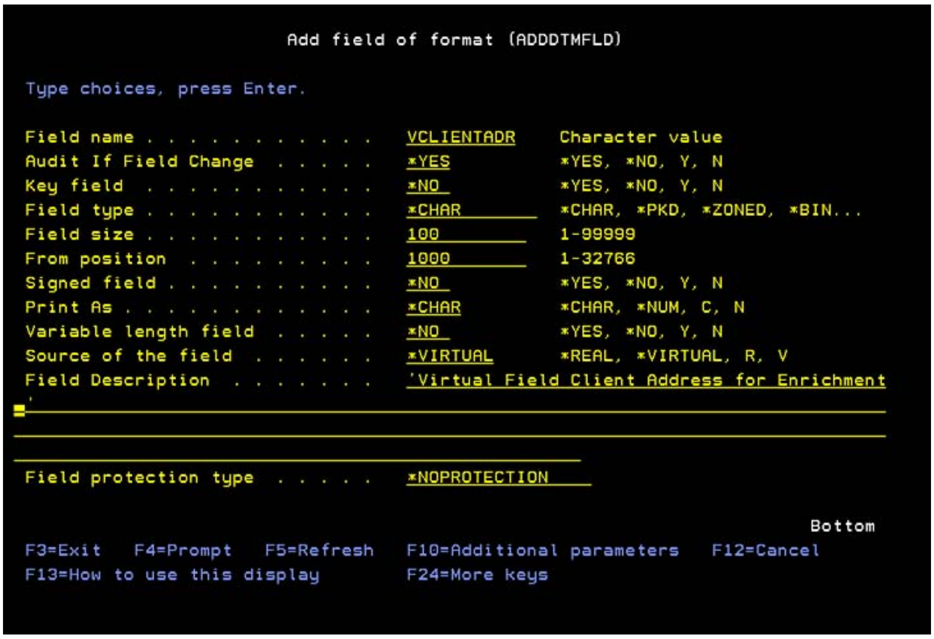


```
(c) Tango/04     Work with monitored files                              8/09/16
                                                                       12:49:54
Type options, press Enter.
  2=Change             4=Delete     5=Display       10=Details
  12=Specify Remote Journal         14=Remove Remote Journal    20=Formats
  30=Enable entry                   31=Disable entry
    System    File        File        Monitor                Latest record date
Opt Name      Name        Library     Status       Enabled     in Historical
__  APOLLO    MOVEMENTS   DMIDEMO     *ACTIVE       *YES     2008/09/16 12:46:02
__  APOLLO    SAMPLE      T4DATAMON   *INACTIVE     *YES










                                                                          Bottom
F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors            F24=More keys
```

Figure 56 – Enter option 20, to work with formats for the MOVEMENTS database

Step 4. Enter option **10**, to work with field details.

Step 5. Enter F6 to add a field, and then enter details for the new field.



```
                        Add field of format (ADDDTMFLD)

 Type choices, press Enter.

 Field name . . . . . . . . . . .   VCLIENTADR    Character value
 Audit If Field Change  . . . . .   *YES          *YES, *NO, Y, N
 Key field  . . . . . . . . . . .   *NO           *YES, *NO, Y, N
 Field type . . . . . . . . . . .   *CHAR         *CHAR, *PKD, *ZONED, *BIN...
 Field size . . . . . . . . . . .   100           1-99999
 From position  . . . . . . . . .   1000          1-32766
 Signed field . . . . . . . . . .   *NO           *YES, *NO, Y, N
 Print As . . . . . . . . . . . .   *CHAR         *CHAR, *NUM, C, N
 Variable length field  . . . . .   *NO           *YES, *NO, Y, N
 Source of the field  . . . . . .   *VIRTUAL      *REAL, *VIRTUAL, R, V
 Field Description  . . . . . . .   'Virtual Field Client Address for Enrichment
'

 Field protection type  . . . . .   *NOPROTECTION

                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
```

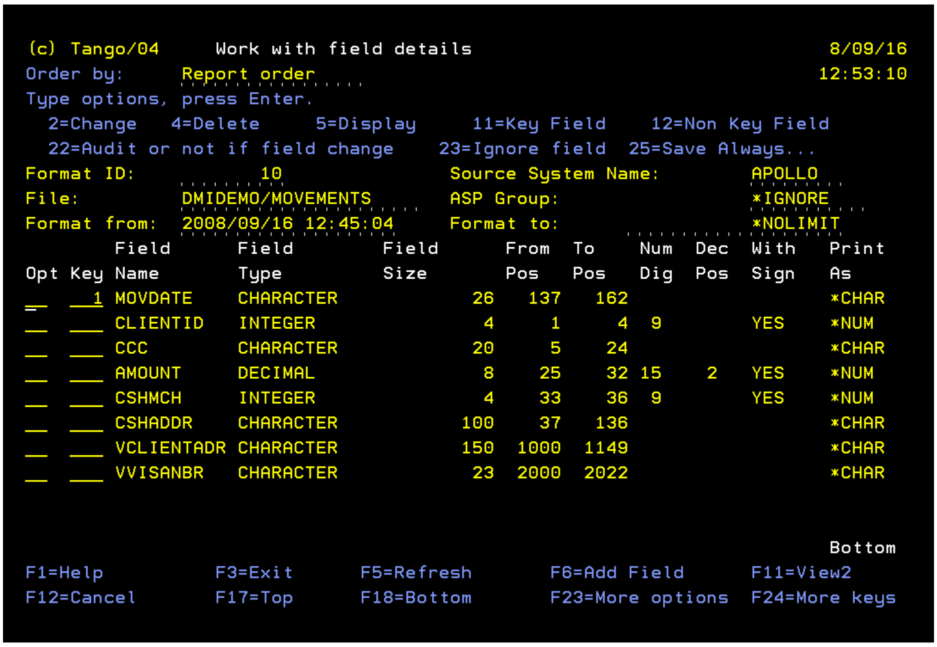
Figure 57 – Enter details for your new virtual field for enrichment, in this example we are adding a field called VCLIENTADR to store address details for a client.

Important

When creating virtual fields, the *From position* data is important because the field cannot overlap other data in the table. Enter a value that is higher than the final position of the other fields in the database.

We recommend allocating enough space to avoid problems with future file format changes, keeping in mind the threshold for the record dimension is about 32Kbytes.

When details have been added press ENTER to return to the field details screen where we can see the new virtual field has been added to the database:



```
(c) Tango/04     Work with field details                               8/09/16
Order by:      Report order                                           12:53:10
Type options, press Enter.
  2=Change   4=Delete    5=Display     11=Key Field    12=Non Key Field
  22=Audit or not if field change    23=Ignore field  25=Save Always...
Format ID:            10               Source System Name:        APOLLO
File:          DMIDEMO/MOVEMENTS       ASP Group:                 *IGNORE
Format from:   2008/09/16 12:45:04     Format to:                 *NOLIMIT
        Field      Field        Field       From  To    Num  Dec  With   Print
Opt Key Name       Type         Size        Pos   Pos   Dig  Pos  Sign   As
 _    1 MOVDATE    CHARACTER           26    137   162                   *CHAR
 __ ___ CLIENTID   INTEGER              4      1     4  9         YES    *NUM
 __ ___ CCC        CHARACTER           20      5    24                   *CHAR
 __ ___ AMOUNT     DECIMAL              8     25    32 15    2    YES    *NUM
 __ ___ CSHMCH     INTEGER              4     33    36  9         YES    *NUM
 __ ___ CSHADDR    CHARACTER          100     37   136                   *CHAR
 __ ___ VCLIENTADR CHARACTER          150   1000  1149                   *CHAR
 __ ___ VVISANBR   CHARACTER           23   2000  2022                   *CHAR


                                                                        Bottom
F1=Help         F3=Exit      F5=Refresh       F6=Add Field      F11=View2
F12=Cancel      F17=Top      F18=Bottom       F23=More options  F24=More keys
```

Figure 58 – The virtual fields VCLIENTADR and VVISANBR have been added to the database. Note the From position data for each field is higher than the previous To position data.

8.4 Detailed example of 360º data enrichment

8.4.1 Scenario

1. Imagine a bank which wants to control its debtors.
2. The bank only wants to monitor the `MOVEMENTS` database. This database stores every single transaction made by its customers, regardless of the method (cash machine, bank branch, Internet, and so forth).
3. The Data Monitor Operator for this bank needs to create an entry configuration for this database, and only this database.
4. It is also important that they can control other external information, which is related to the transaction data from this database. This information is stored on two other databases:
    - `DEBTORS`
    - `CLIENTS`
5. These databases don't have to be monitored, but the information does need to be accessed in order to be used for controls and to enrich any reports that will be produced.
6. The adequate conditions needed to check these external databases will be created using the new ALEV functions in our expressions.
7. The bank also wants to create a log of all the transactions for debtors only, in another external database. This log can be on the local machine or even on a remote machine.

8.4.2 Databases

The following databases are used in our example:

- `DMIDEMO/MOVEMENTS`: this is the database monitored by Data Monitor where every transaction of the bank customers will be saved.
- `DMIDEMO/CLIENTS`: bank customer database.
- `DMIDEMO/DEBTORS`: database containing details of the bank debtors.
- `DMIDEMO/TRACKDBT`: database storing details of the transactions generated by `DEBTORS` who match the specified conditions we are going to configure in our expressions.

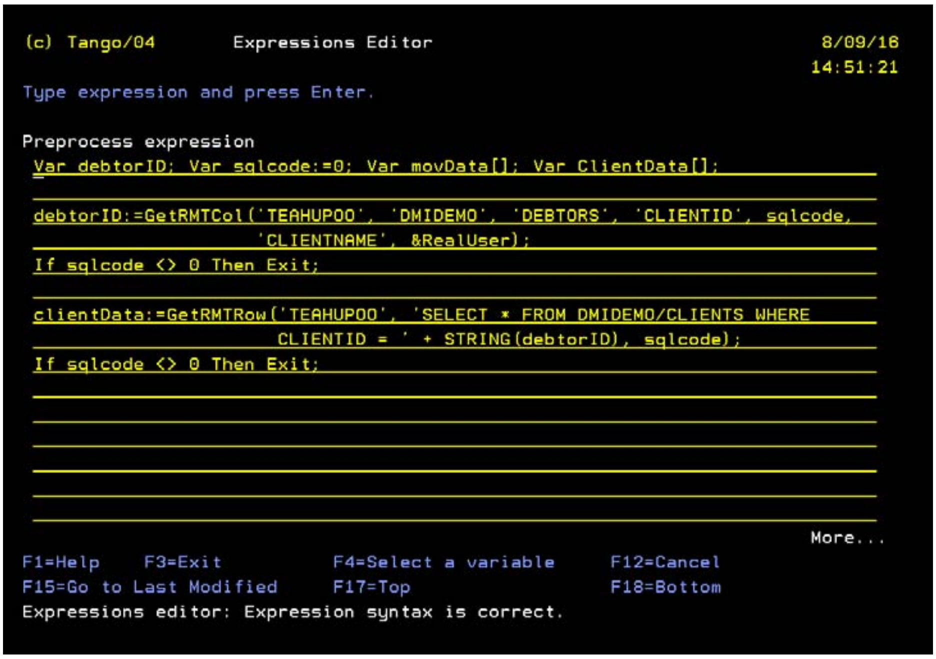
8.4.3 What we will do

In our example, we are going to:

- Check if a customer, responsible for a specific transaction, is in our database of debtors. This database (`DEBTORS`) exists on a remote machine.
- Then we are going to get detailed information regarding this customer (debtor).
- Finally we are going to insert a new record in our transaction log database for transactions of debtors.

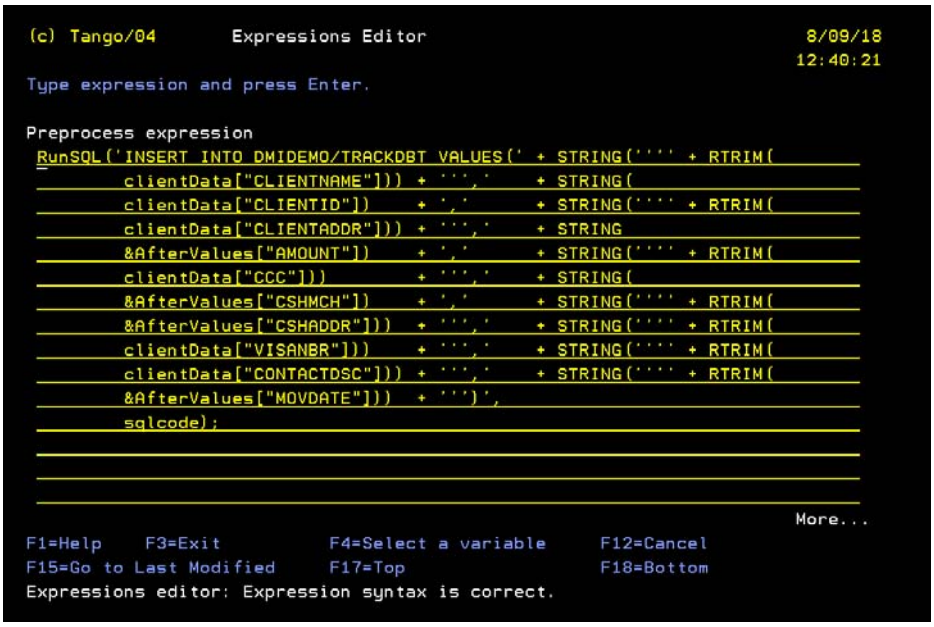
8.4.4 Script

We will use the following script to achieve all of the above.



```
(c)  Tango/04        Expressions Editor                          8/09/16
                                                                14:51:21
Type expression and press Enter.

Preprocess expression
 Var debtorID; Var sqlcode:=0; Var movData[]; Var ClientData[];

 debtorID:=GetRMTCol('TEAHUPOO', 'DMIDEMO', 'DEBTORS', 'CLIENTID', sqlcode,
                    'CLIENTNAME', &RealUser);
 If sqlcode <> 0 Then Exit;

 clientData:=GetRMTRow('TEAHUPOO', 'SELECT * FROM DMIDEMO/CLIENTS WHERE
                      CLIENTID = ' + STRING(debtorID), sqlcode);
 If sqlcode <> 0 Then Exit;






                                                                 More...
F1=Help    F3=Exit          F4=Select a variable     F12=Cancel
F15=Go to Last Modified     F17=Top                  F18=Bottom
Expressions editor: Expression syntax is correct.
```

Figure 59 – Preprocess script used in our example, in the Expressions Editor - first screen



```
(c)  Tango/04        Expressions Editor                          8/09/18
                                                                12:40:21
Type expression and press Enter.

Preprocess expression
 RunSQL('INSERT INTO DMIDEMO/TRACKDBT VALUES(' + STRING('''' + RTRIM(
         clientData["CLIENTNAME"])) + ''','     + STRING(
         clientData["CLIENTID"])    + ','      + STRING('''' + RTRIM(
         clientData["CLIENTADDR"])) + ''','     + STRING
         &AfterValues["AMOUNT"])    + ','      + STRING('''' + RTRIM(
         clientData["CCC"]))        + ''','     + STRING(
         &AfterValues["CSHMCH"])    + ','      + STRING('''' + RTRIM(
         &AfterValues["CSHADDR"]))  + ''','     + STRING('''' + RTRIM(
         clientData["VISANBR"]))    + ''','     + STRING('''' + RTRIM(
         clientData["CONTACTDSC"])) + ''','     + STRING('''' + RTRIM(
         &AfterValues["MOVDATE"]))  + ''')',
         sqlcode);




                                                                 More...
F1=Help    F3=Exit          F4=Select a variable     F12=Cancel
F15=Go to Last Modified     F17=Top                  F18=Bottom
Expressions editor: Expression syntax is correct.
```

Figure 60 – Preprocess script used in our example, in the Expressions Editor - second screen

First of all we need to declare some variables. It is important to initialize the sqlcode to let ALEV know we are going to use an `INTEGER` type for this variable. To find out more please refer to the *ALEV Reference Manual*.

```
Var debtorID; Var sqlcode:=0; Var movData[]; Var
clientData[];
```

Note

The Var sqlcode must be an integer.

Next we need to know if the user for this transaction is in our debtor database. This could be done as easy as writing the following lines in our expressions:

```
        debtorID:=GetRMTCol('TEAHUPOO',    'DMIDEMO',    'DEBTORS',
        'CLIENTID', sqlcode, 'CLIENTNAME', &RealUser);

        If sqlcode <> 0 Then Exit;
```

Notice we are using the SQLCODE of our functions to check if the function has returned the information correctly. With this function we are going to retrieve the debtor identifier and automatically the function returns the value into the debtorID variable. From now on, the debtorID could be treated as an integer in our script.

Now, we need to get detailed customer information from a remote machine. This is done with the following lines:

```
        clientData:=GetRMTRow('TEAHUPOO',  'SELECT  *  FROM  DMIDEMO/
        CLIENTS WHERE CLIENTID = ' + STRING(debtorID), sqlcode);

        If sqlcode <> 0 Then Exit;
```

We want to log detailed transaction information for this debtor in another database called TRACKDBT, which is not monitored. This is done with the following lines:

```
RunSQL('INSERT  INTO  DMIDEMO/TRACKDBT  VALUES('  +  STRING(''''  +
        RTRIM(

        clientData["CLIENTNAME"]))   +''',' + STRING(

        clientData["CLIENTID"])      + ','  + STRING('''' +  RTRIM(

        clientData["CLIENTADDR"]))   + ''','+ STRING

        &AfterValues["AMOUNT"])      + ','  + STRING('''' +  RTRIM(

        clientData["CCC"]))          + ''','+ STRING(

        &AfterValues["CSHMCH"])      + ','  + STRING('''' +  RTRIM(

        &AfterValues["CSHADDR"]))    + ''','+ STRING('''' +  RTRIM(

        clientData["VISANBR"]))      + ''','+ STRING('''' +  RTRIM(

        clientData["CONTACTDSC"]))   + ''','+ STRING('''' +  RTRIM(

        &AfterValues["MOVDATE"]))    +''')',

        sqlcode);
```

8.4.5 Enriching our monitored database

Now, we can make this even more interesting by enriching our monitored database with this external information. All we need to do is simply create some new virtual fields in the current format for a specified database from the Data Monitor for Files user interface, and then work with the functions.

```
        &OperationValues[“VIRTUALFIELD”]:=      <new       external
        information retrieved>.
```

We will create two virtual fields:

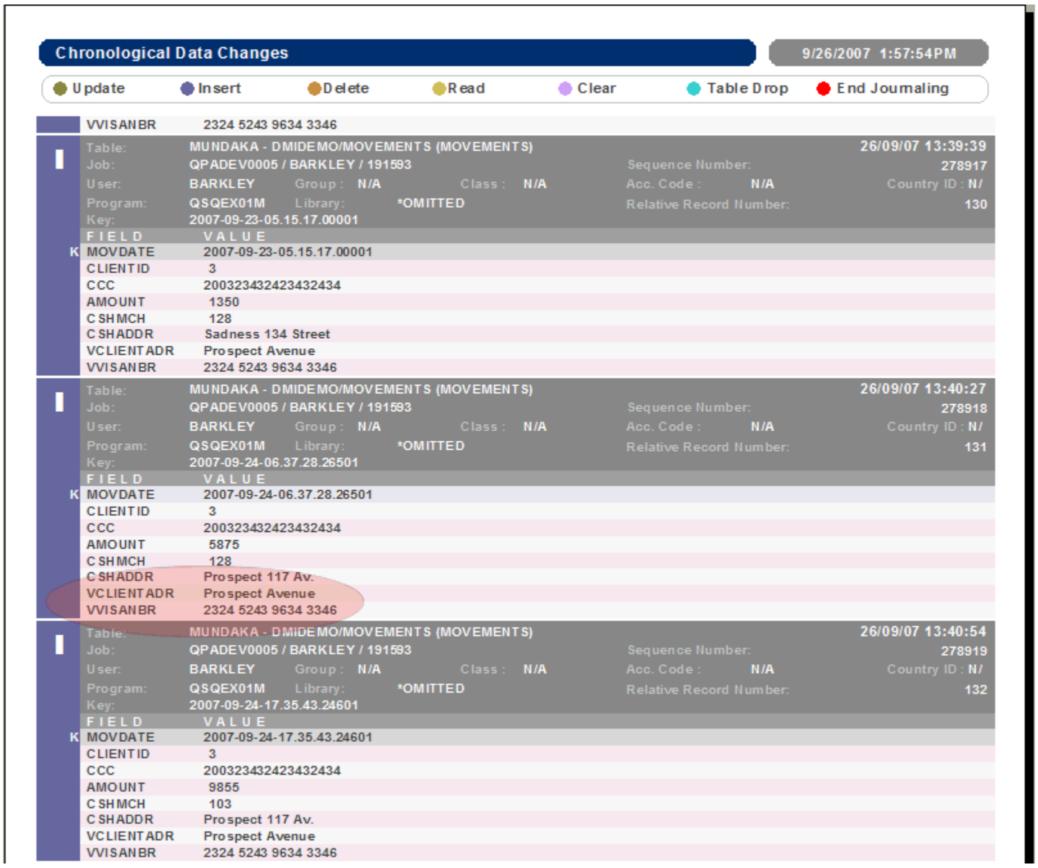
- `VCLIENTADR`: to store customer address information
- `VVISANBR`: to store VISA number details

For further details on how to create virtual fields *see section 8.3 - Create a virtual field for enrichment on page 70.*

Once you have created your new virtual fields you can write the following lines in your preprocess expression in order to save these mentioned values:

```
&OperationValues[“VCLIENTADR”]:=clientData[“CLIENTADDR”];

&OperationValues[“VVISANBR”]:=clientData[“VISANBR”];
```

Finally, we can run a report from Reports bringing all the data from the various databases together:



```
      VVISANBR        2324 5243 9634 3346
 I    Table:          MUNDAKA - DMIDEMO/MOVEMENTS (MOVEMENTS)                                      26/09/07 13:39:39
      Job:            QPADEV0005 / BARKLEY / 191593                    Sequence Number:                     278917
      User:           BARKLEY    Group :  N/A      Class :   N/A       Acc. Code :       N/A          Country ID : N/
      Program:        QSQEX01M   Library:       *OMITTED               Relative Record Number:                 130
      Key:            2007-09-23-05.15.17.00001
      FIELD           VALUE
    K MOVDATE         2007-09-23-05.15.17.00001
      CLIENTID         3
      CCC             200323432423432434
      AMOUNT           1350
      CSHMCH           128
      CSHADDR         Sadness 134 Street
      VCLIENTADR      Prospect Avenue
      VVISANBR        2324 5243 9634 3346
 I    Table:          MUNDAKA - DMIDEMO/MOVEMENTS (MOVEMENTS)                                      26/09/07 13:40:27
      Job:            QPADEV0005 / BARKLEY / 191593                    Sequence Number:                     278918
      User:           BARKLEY    Group :  N/A      Class :   N/A       Acc. Code :       N/A          Country ID : N/
      Program:        QSQEX01M   Library:       *OMITTED               Relative Record Number:                 131
      Key:            2007-09-24-06.37.28.26501
      FIELD           VALUE
    K MOVDATE         2007-09-24-06.37.28.26501
      CLIENTID         3
      CCC             200323432423432434
      AMOUNT           5875
      CSHMCH           128
      CSHADDR         Prospect 117 Av.
      VCLIENTADR      Prospect Avenue
      VVISANBR        2324 5243 9634 3346
 I    Table:          MUNDAKA - DMIDEMO/MOVEMENTS (MOVEMENTS)                                      26/09/07 13:40:54
      Job:            QPADEV0005 / BARKLEY / 191593                    Sequence Number:                     278919
      User:           BARKLEY    Group :  N/A      Class :   N/A       Acc. Code :       N/A          Country ID : N/
      Program:        QSQEX01M   Library:       *OMITTED               Relative Record Number:                 132
      Key:            2007-09-24-17.35.43.24601
      FIELD           VALUE
    K MOVDATE         2007-09-24-17.35.43.24601
      CLIENTID         3
      CCC             200323432423432434
      AMOUNT           9855
      CSHMCH           103
      CSHADDR         Prospect 117 Av.
      VCLIENTADR      Prospect Avenue
      VVISANBR        2324 5243 9634 3346
```

Figure 61 – Report displaying linked data from several databases. We can see that the virtual fields we created for Client address and VISA number have been included in the report.

8.5 How to work with members

It is possible to work with different members when working with files. By default, you will be working with the first member (`*FIRST`) if no member is specified; however it is possible to specify a different member from the default.

When using the new ALEV functions for enrichment there are no parameters which can be used to set the member. We can however use a `CREATE ALIAS SQL` statement. With this command you can specify an `ALIAS` for a specific member in a file and then you can work with this `ALIAS` as if it was a normal file.

For example, imagine you have a file, `FILEA` with two members: `MBR1` and `MBR2`. If you want to work with `MBR1`, then there is no problem, because by default you will be working with it. But if you want to work with `MBR2` then you will need to specify this as the member to work with.

You can solve the situation by running the following SQL statement (with interactive SQL for example):

```
‘CREATE ALIAS MYLIBRARY/MYMEMBER FOR MYLIBRARY/FILEA (MBR2)
```

You will only need to run this SQL statement once.

Then you can use our SQL functions as usual by referring to `MYLIBRARY/MYMEMBER`. For example:

```
RecordData[]:=GetRow(‘SELECT   *   FROM   MYLIBRARY/MYMEMBER
WHERE FIELDA = <valueA>’, sqlcode);

If sqlcode <> 0 Then Exit;
```

Chapter 9 Performance Considerations

Although Data Monitor for iSeries for Sensitive Files reads transactions from journals, it works at file level, not at journal level. So, if the same journal is used for several journaled files, Data Monitor only processes transactions for the sensitive files that are specified, and not those of the entire journal. Likewise, if Data Monitor for iSeries for Sensitive Users reads transactions from journals, it only processes transactions for the sensitive users that are specified and not those of the entire journal, as it works at user level.

To further narrow down the information monitored by Data Monitor you can specify the type of transaction to process. For example, if you are only interested in the modifications, you can set the inserts and deletes to `*NO` and only the update transactions will be processed.

Data monitor can also audit Read transactions. To prevent overburdening the system by auditing all read transactions, the default value for auditing read transactions is `*NO`. The user can change this to `*YES` on a file-by-file basis. We recommend you audit reads only on the most sensitive files, for example your `PAYROLL` or `SALARY` file.

In addition, you can refine the monitoring process down to field level. By defining sensitive fields only transactions related to these fields will be listed.

Important
It is very important to configure Data Monitor properly to decrease the CPU time used by the monitors. A bad configuration could lead to unnecessarily processing millions of transactions a day, thus wasting valuable system resources.

Furthermore Data Monitor can read from journals and journal receivers at any time. Depending on the number of transactions to process and your system resources you can run Data Monitor during production hours. Or you may decide it makes more sense to run Data Monitor outside working hours, thereby reducing the load on your production system. Data Monitor offers this flexibility.

Finally, Data Monitor also has Remote Journal Support. You can use Remote Journals to run Data Monitor on a different system from your production machine. OS/400 (or i5/OS) takes care of journal replication from the production system to system with the remote journal. That way impact on your production system performance is near zero!

9.1 Data Monitor for iSeries Files

Only in Data Monitor for iSeries Files can you activate the trigger to monitor read transactions on a file. The generated read transactions are then stored in the associated journal where they are available for use in Data Monitor for iSeries Users and Express. As mentioned earlier in this document auditing reads may generate a large number of transactions and we recommend you only activate triggers on files where it is really necessary.

9.2 Data Monitor for iSeries Users

Data Monitor for iSeries users might work with many journals, which implies working with many files at the same time, potentially resulting in a very large number of transactions to store. Large amounts of read transactions in the system further compound the volume of data to process.

We strongly recommend you reduce the volume of data to store by using the filters

- Monitoring options (option F15 from Work with Monitored Users)
- Store transaction if (option F14 from Work with Monitored Users)

We further suggest you carefully define what actions and alerts to audit (remember auditing reads for users may cause large overhead).

Finally, we suggest you avoid duplication of your monitoring efforts, as described in *Appendix L: Best Practices on page 153*.

9.3 Data Monitor for iSeries Express

Data Monitor for iSeries Express allows you to create once off auditing data for reporting purposes. You can narrow down your audit for specific periods of time. Even though you can run an audit for a short length of time, you still need to take the volume of read transactions on your system into account for performance considerations.

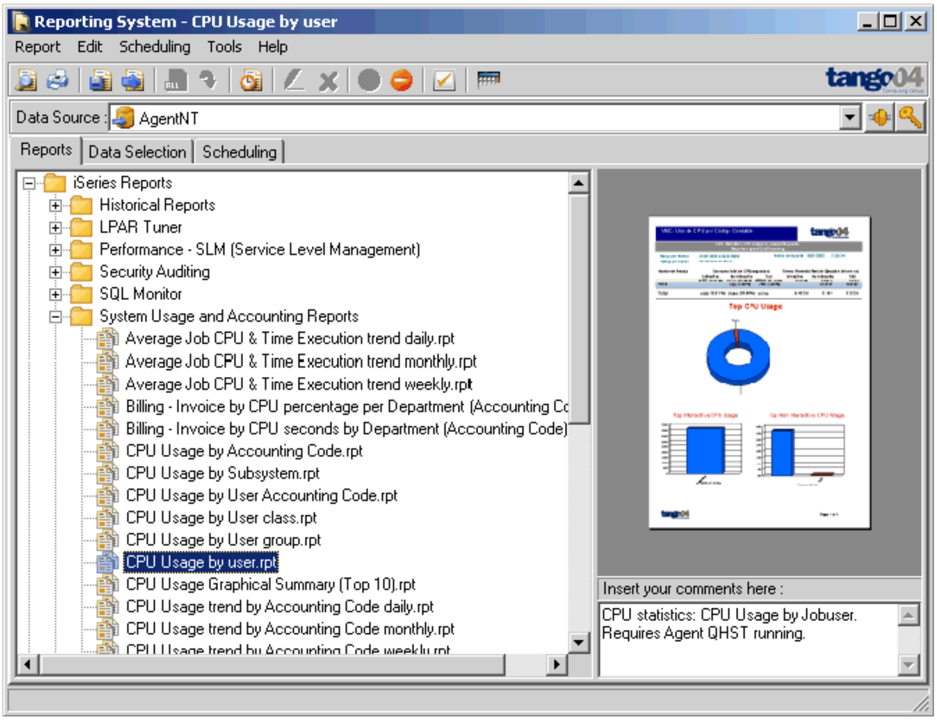
Data Monitor for iSeries Express is very flexible and allows you many parameters to help narrow down your audit to the most important elements. We recommend you make full use of the available options to reduce the volume of data to process. For details *see Chapter 5 - Data Monitor for iSeries Express on page 53.*

Chapter 10
Data Monitor Reports

You can use either the included VISUAL Message Center Reports or create your own internal database queries. Reports is a very flexible means of obtaining information, since it allows you to define your selection in an intuitive manner, using an easy to use GUI, choosing from a wide range of selection parameters. For advanced users, there is even more flexibility, since the selection criteria can be expanded to any SQL selection clause. There are more than 150 different predefined reports that cover Operations, Service Level Management, Job Accounting, Security and other areas, in detailed, summary, and graphical formats; and that number is continuously growing.

Reports can be exported into a variety of formats, including PDF (Adobe Acrobat), plain text, HTML, Windows Word document format, etc. A complete list of available formats is included at the end of this chapter. Reports can be sent by email right from Reports, too.

Reports can also be scheduled. For instance, you can automatically feed your corporate portal with auditing reports every night (or at any interval).



Data Source : AgentNT

- iSeries Reports
  - Historical Reports
  - LPAR Tuner
  - Performance - SLM (Service Level Management)
  - Security Auditing
  - SQL Monitor
  - System Usage and Accounting Reports
    - Average Job CPU & Time Execution trend daily.rpt
    - Average Job CPU & Time Execution trend monthly.rpt
    - Average Job CPU & Time Execution trend weekly.rpt
    - Billing - Invoice by CPU percentage per Department (Accounting Cc
    - Billing - Invoice by CPU seconds by Department (Accounting Code)
    - CPU Usage by Accounting Code.rpt
    - CPU Usage by Subsystem.rpt
    - CPU Usage by User Accounting Code.rpt
    - CPU Usage by User class.rpt
    - CPU Usage by User group.rpt
    - CPU Usage by user.rpt
    - CPU Usage Graphical Summary (Top 10).rpt
    - CPU Usage trend by Accounting Code daily.rpt
    - CPU Usage trend by Accounting Code monthly.rpt
    - CPU Usage trend by Accounting Code weekly.rpt

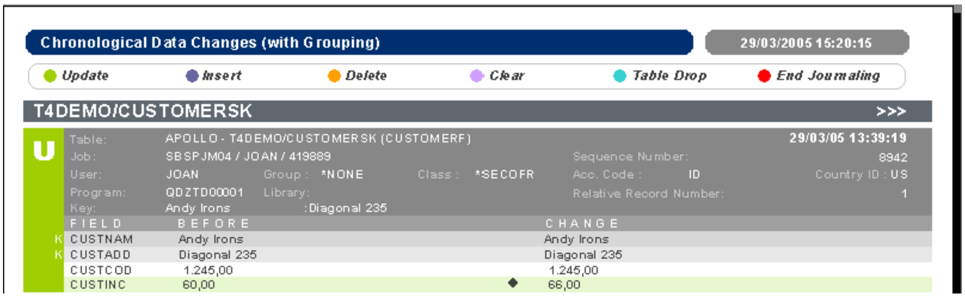
Insert your comments here :

CPU statistics: CPU Usage by Jobuser.
Requires Agent QHST running.

Figure 62 – Reports contains over 150 reports to analyze the operational health and security of your IT infrastructure. Customized reports can be created easily.

10.1 Predefined Data Monitor Reports

Data Monitor is fully integrated with Reports for creating high level auditing reports as well as field level detail reports. There are several predefined reports that you can use immediately from Reports to extract data from Data Monitor’s Historical Transaction Log. One set of reports is located in the Data Monitor folder under the iSeries Reports branch of the Reports tree.



Chronological Data Changes (with Grouping) 29/03/2005 15:20:15

● Update ● Insert ● Delete ● Clear ● Table Drop ● End Journaling

T4DEMO/CUSTOMERSK >>>

U

Table: APOLLO - T4DEMO/CUSTOMERSK (CUSTOMERF) 29/03/05 13:39:19

Job: SBSPJM04 / JOAN / 419889 Sequence Number: 8942

User: JOAN Group : *NONE Class : *SECOFR Acc. Code : ID Country ID : US

Program: QDZTD00001 Library: Relative Record Number: 1

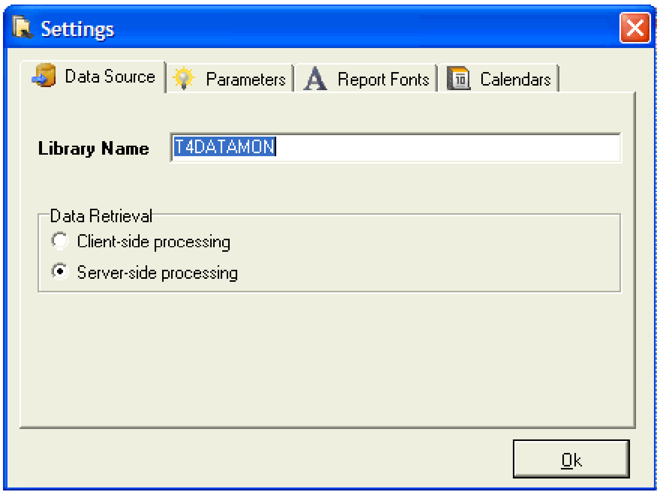
Key: Andy Irons :Diagonal 235

	FIELD	BEFORE		CHANGE
K	CUSTNAM	Andy Irons		Andy Irons
K	CUSTADD	Diagonal 235		Diagonal 235
	CUSTCOD	1.245,00		1.245,00
	CUSTINC	60,00	◆	66,00

Figure 63 – Using Reports you can easily view, print, email, or transform into HTML listings of changes made to sensitive files.

Note

Remember that Data Monitor for Sensitive Users and Data Monitor Express can be configured to store the historical transaction log in different libraries from default `T4DATAMON` library. To access this data with Data Monitor iSeries Reports you need to set this library in the report. To set the new library, open the Tools menu and select Settings. Here enter the Library you wish to use for the Data Monitor iSeries Reports.



Library Name: T4DATAMON

Data Retrieval

○ Client-side processing

● Server-side processing

Figure 64 – Enter the library you wish to use for Data Monitor iSeries reports

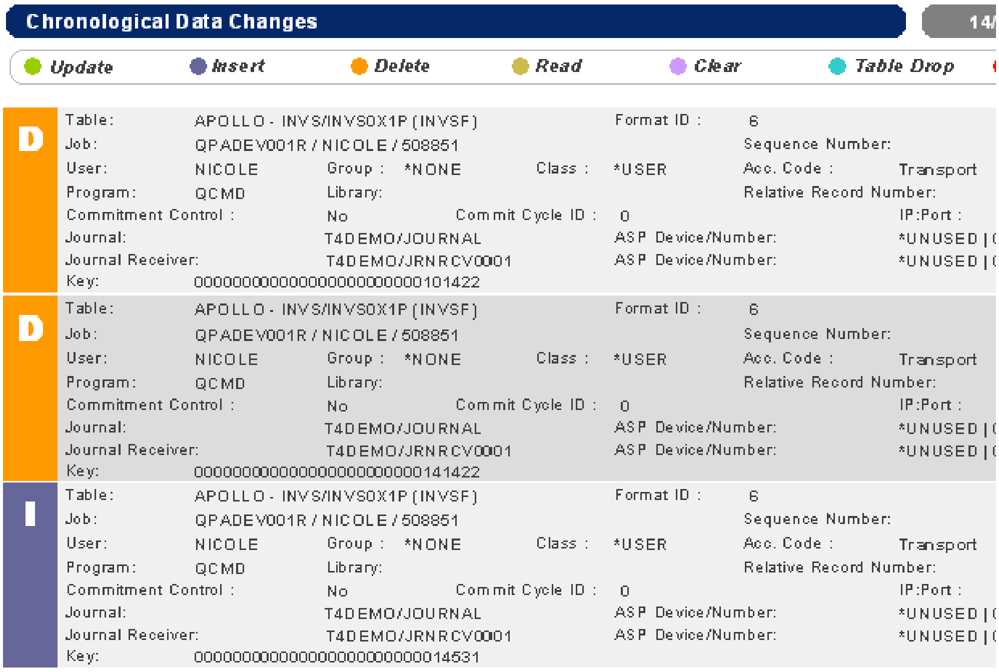
The reports you will find there are: Chronological Data Changes (Transaction Header Information Only), Chronological Data Changes, Chronological Data Changes (With Grouping) and Total Transactions Summary.

10.1.1 Chronological Data Changes (Transaction Header Information Only)

The transaction header report gives you quick insight into the data changes to files and is faster to run than the other two more detailed reports. This report is particularly useful for checking access and access type for a specific database.

For instance use this report to identify if there was any suspicious access by company employees. If you already suspect a particular user, you can narrow down the report even further. Let’s say that you find a particular user who accessed a sensitive file on Friday afternoon. You can now use any of the other, more complete, Data Monitor reports to get more detail, such as what file and specifically what data did this person access.

For example: let's check the invoices database `INVS/INVS0X1P` on Apollo to see which users of `*USER` class, that belong to the accounting code Transport, have accessed the database in the last quarter.



```
Chronological Data Changes                                                        14/

● Update        ● Insert        ● Delete        ● Read        ● Clear        ● Table Drop

D  Table:        APOLLO - INVS/INVS0X1P (INVSF)                 Format ID :     6
   Job:          QPADEV001R / NICOLE / 508851                                   Sequence Number:
   User:         NICOLE        Group :   *NONE       Class :   *USER           Acc. Code :          Transport
   Program:      QCMD          Library:                                        Relative Record Number:
   Commitment Control :        No             Commit Cycle ID :   0                                 IP:Port :
   Journal:                    T4DEMO/JOURNAL                   ASP Device/Number:                  *UNUSED |
   Journal Receiver:           T4DEMO/JRNRCV0001                ASP Device/Number:                  *UNUSED |
   Key:          000000000000000000000000101422

D  Table:        APOLLO - INVS/INVS0X1P (INVSF)                 Format ID :     6
   Job:          QPADEV001R / NICOLE / 508851                                   Sequence Number:
   User:         NICOLE        Group :   *NONE       Class :   *USER           Acc. Code :          Transport
   Program:      QCMD          Library:                                        Relative Record Number:
   Commitment Control :        No             Commit Cycle ID :   0                                 IP:Port :
   Journal:                    T4DEMO/JOURNAL                   ASP Device/Number:                  *UNUSED |
   Journal Receiver:           T4DEMO/JRNRCV0001                ASP Device/Number:                  *UNUSED |
   Key:          000000000000000000000000141422

I  Table:        APOLLO - INVS/INVS0X1P (INVSF)                 Format ID :     6
   Job:          QPADEV001R / NICOLE / 508851                                   Sequence Number:
   User:         NICOLE        Group :   *NONE       Class :   *USER           Acc. Code :          Transport
   Program:      QCMD          Library:                                        Relative Record Number:
   Commitment Control :        No             Commit Cycle ID :   0                                 IP:Port :
   Journal:                    T4DEMO/JOURNAL                   ASP Device/Number:                  *UNUSED |
   Journal Receiver:           T4DEMO/JRNRCV0001                ASP Device/Number:                  *UNUSED |
   Key:          000000000000000000000000014531
```

Figure 65 – Excerpt from Chronological Data Changes (Transaction Header Information Only)

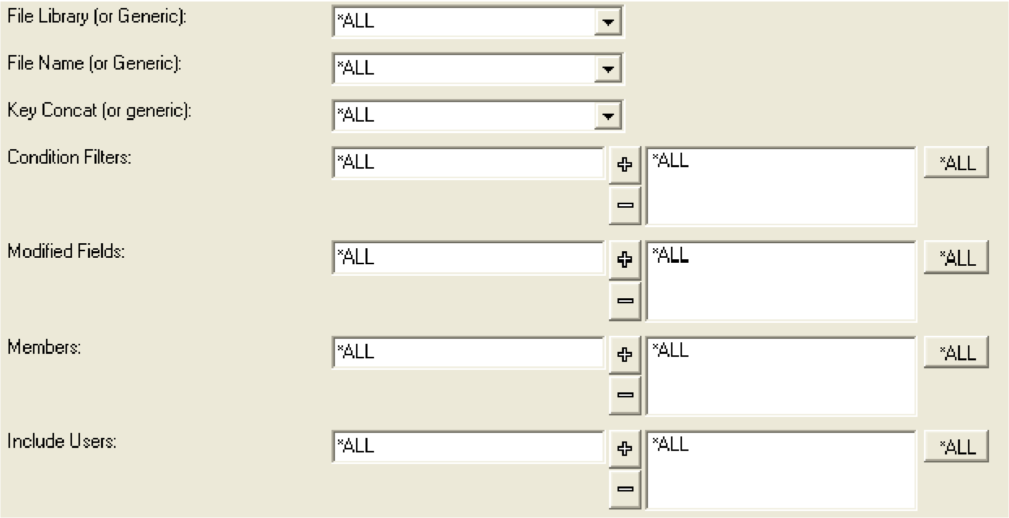
It returns a list containing basic information about the individual transactions. Header details include table, job, user, program, journal, journal receiver, ASP device, key, and more.

The last page of the report presents a summary of changes per operation type.

10.1.2 Chronological Data Changes

The Chronological Data Changes report offers the most detailed level of information. It shows a chronological list of changes made to files.

Special features of this report allow you to use Condition Filters (described in the next report) and select which Modified Fields you want to include to refine your report.



File Library (or Generic): *ALL

File Name (or Generic): *ALL

Key Concat (or generic): *ALL

Condition Filters: *ALL *ALL

Modified Fields: *ALL *ALL

Members: *ALL *ALL

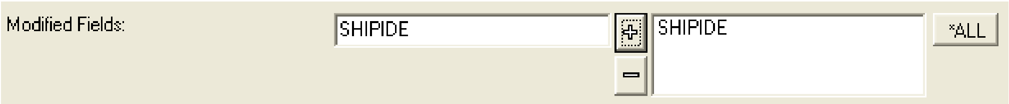
Include Users: *ALL *ALL

Figure 66 – Special selection parameters Condition Filters and Modified Fields

The default setting for the *Modified Field* parameter is `*ALL`, which means that transactions will be added for any changed fields. Selecting one or more specific fields so that your report only shows those transactions where a particular field was changed.

For example a logistics company needs to know what Operators made changes to the ID of the shipping documents `SHIPIDE` on database `SHIPMENTS/SHIP0X1P`.

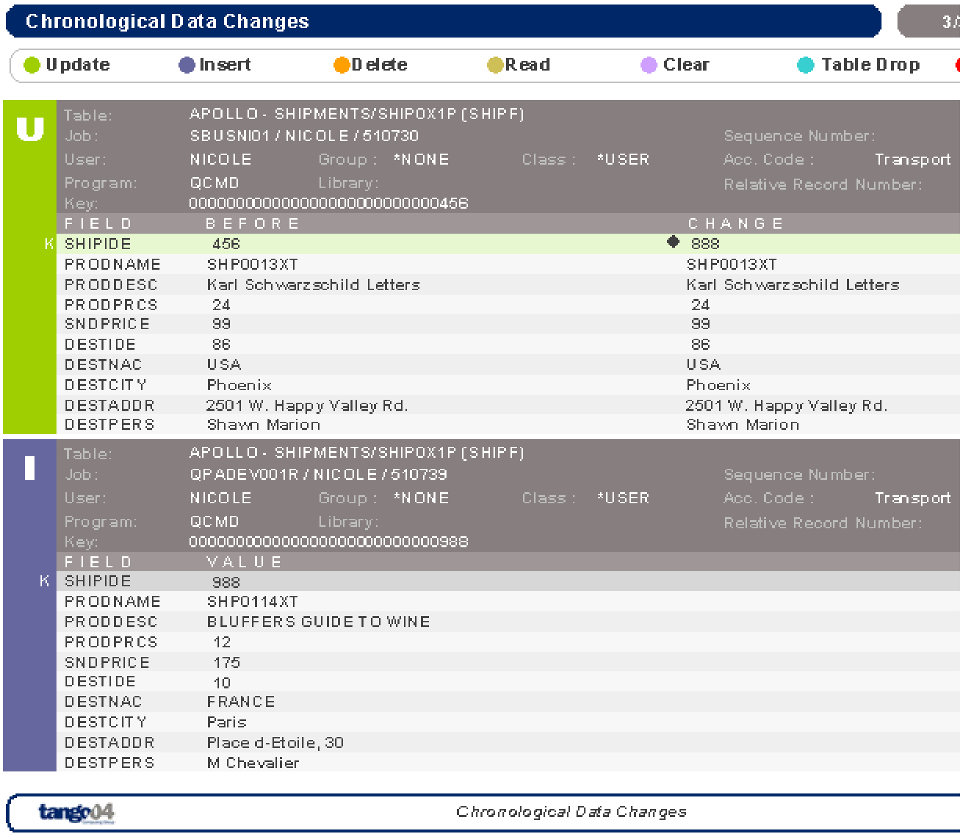
For this example we enter `SHIPIDE` in the Modified Fields parameter



Modified Fields: SHIPIDE SHIPIDE

Figure 67 – Example of Modified Fields

The next image shows an excerpt of the resulting report



Chronological Data Changes

● Update ● Insert ● Delete ● Read ● Clear ● Table Drop

U

Table: APOLLO - SHIPMENTS/SHIPOX1P (SHIPF)
Job: SBUSNI01 / NICOLE / 510730 Sequence Number:
User: NICOLE Group : *NONE Class : *USER Acc. Code : Transport
Program: QCMD Library: Relative Record Number:
Key: 00000000000000000000000000000456

	FIELD	BEFORE		CHANGE
K	SHIPIDE	456	◆	888
	PRODNAME	SHP0013XT		SHP0013XT
	PRODDESC	Karl Schwarzschild Letters		Karl Schwarzschild Letters
	PRODPRCS	24		24
	SNDPRICE	99		99
	DESTIDE	86		86
	DESTNAC	USA		USA
	DESTCITY	Phoenix		Phoenix
	DESTADDR	2501 W. Happy Valley Rd.		2501 W. Happy Valley Rd.
	DESTPERS	Shawn Marion		Shawn Marion

I

Table: APOLLO - SHIPMENTS/SHIPOX1P (SHIPF)
Job: QPADEV001R / NICOLE / 510739 Sequence Number:
User: NICOLE Group : *NONE Class : *USER Acc. Code : Transport
Program: QCMD Library: Relative Record Number:
Key: 00000000000000000000000000000988

	FIELD	VALUE
K	SHIPIDE	988
	PRODNAME	SHP0114XT
	PRODDESC	BLUFFERS GUIDE TO WINE
	PRODPRCS	12
	SNDPRICE	175
	DESTIDE	10
	DESTNAC	FRANCE
	DESTCITY	Paris
	DESTADDR	Place d-Etoile, 30
	DESTPERS	M Chevalier

tango04 Chronological Data Changes

Figure 68 – Excerpt from Chronological Data Changes

Besides header data such as table, job, user, and program, it shows field-level details of the transactions. For example, for update records in the customer table the field details include user name and address, customer code and the changes that were made to them. In our example the new SHIPIDE number is marked with a black dot.

The last page shows a summary of number of changes per operation type, both in table form and as a bar chart.

10.1.3 Chronological Data Changes (With Grouping)

The With Grouping report offers the same detail as the Chronological Data Changes report, but allows you to group the results according to System/Filename, User Name, Program name or Concatenated Key making the changes. This helps identify specific problem areas.

Special features of this report allow you to use Condition Filters and select which modified fields you want to include to refine your report. See the previous report description for details regarding modified fields. The condition filter parameter is described here.

You can use the *Condition Filter* to narrow down your results based on field contents, by adding the conditions to an inclusion list. For example:

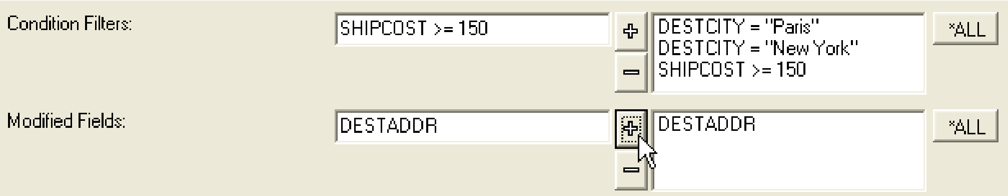
- FNAME = “John”
- CODTAG LIKE “CPF%”
- CITY <> “New York”
- SALES >= 5000

- YEAR < 2000

The accepted comparison signs are `=`, `>`, `<`, `>=` ,`<=` ,`<>` and `LIKE`. If the value is a string the value should be placed in double parenthesis (").

For example: let's say that a logistics company needs to control changes made to the destination address field `DESTADDR` (modified field) in the database `SHIPMENTS/SHIP0X1P` this year by operator, but only for packages shipped to Paris and New York or where the value of the shipping costs exceeded 150 Euros (condition filters).

For this example we enter the following condition filters and modified fields:



Condition Filters: SHIPCOST >= 150

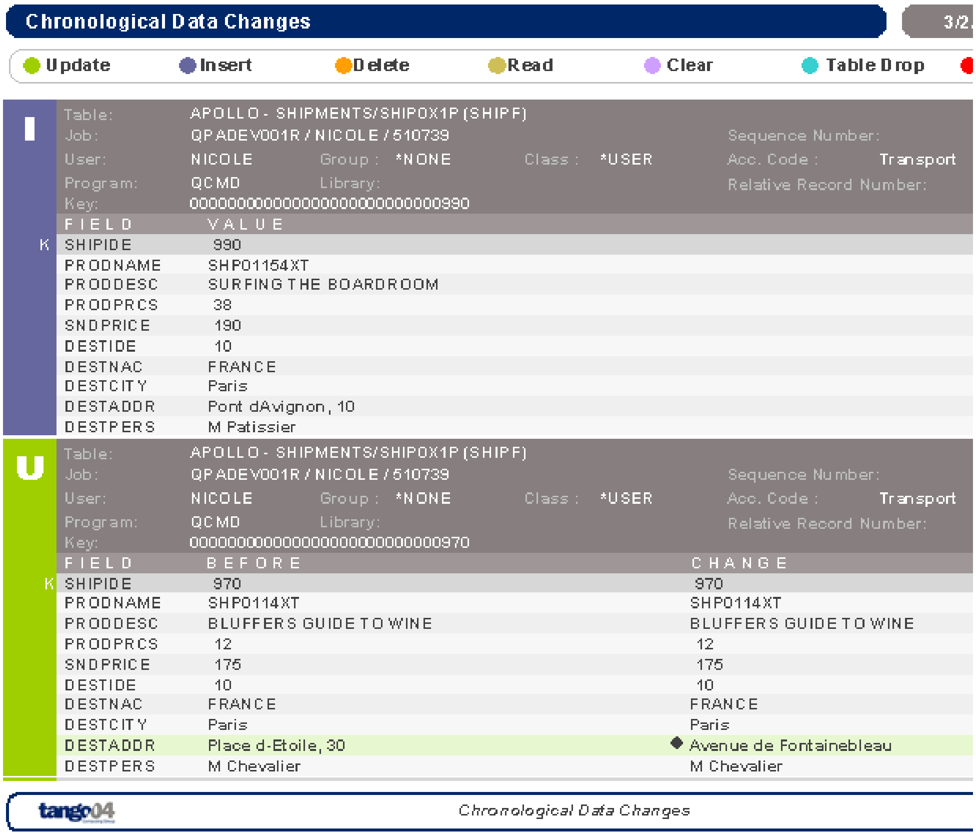
DESTCITY = "Paris"
DESTCITY = "New York"
SHIPCOST >= 150

Modified Fields: DESTADDR

DESTADDR

Figure 69 – Example of Condition Filters and Modified Fields

The following image shows an excerpt of the resulting report.



Chronological Data Changes

Table:	APOLLO - SHIPMENTS/SHIP0X1P (SHIPF)					
Job:	QPADEV001R / NICOLE / 510739				Sequence Number:	
User:	NICOLE	Group :	*NONE	Class : *USER	Acc. Code :	Transport
Program:	QCMD	Library:			Relative Record Number:	
Key:	000000000000000000000000000990					

	FIELD	VALUE
K	SHIPIDE	990
	PRODNAME	SHP01154XT
	PRODDESC	SURFING THE BOARDROOM
	PRODPRCS	38
	SNDPRICE	190
	DESTIDE	10
	DESTNAC	FRANCE
	DESTCITY	Paris
	DESTADDR	Pont dAvignon, 10
	DESTPERS	M Patissier

Table:	APOLLO - SHIPMENTS/SHIP0X1P (SHIPF)					
Job:	QPADEV001R / NICOLE / 510739				Sequence Number:	
User:	NICOLE	Group :	*NONE	Class : *USER	Acc. Code :	Transport
Program:	QCMD	Library:			Relative Record Number:	
Key:	000000000000000000000000000970					

	FIELD	BEFORE		CHANGE
K	SHIPIDE	970		970
	PRODNAME	SHP0114XT		SHP0114XT
	PRODDESC	BLUFFERS GUIDE TO WINE		BLUFFERS GUIDE TO WINE
	PRODPRCS	12		12
	SNDPRICE	175		175
	DESTIDE	10		10
	DESTNAC	FRANCE		FRANCE
	DESTCITY	Paris		Paris
	DESTADDR	Place d-Etoile, 30	◆	Avenue de Fontainebleau
	DESTPERS	M Chevalier		M Chevalier

Chronological Data Changes

Figure 70 – Excerpt from Chronological Data Changes (With Grouping), grouped by user

For each User (operator) the report lists the transaction header and details, and shows a summary of the operation type for each user. In our example the changed `DESTADDR` field is marked with a black dot.

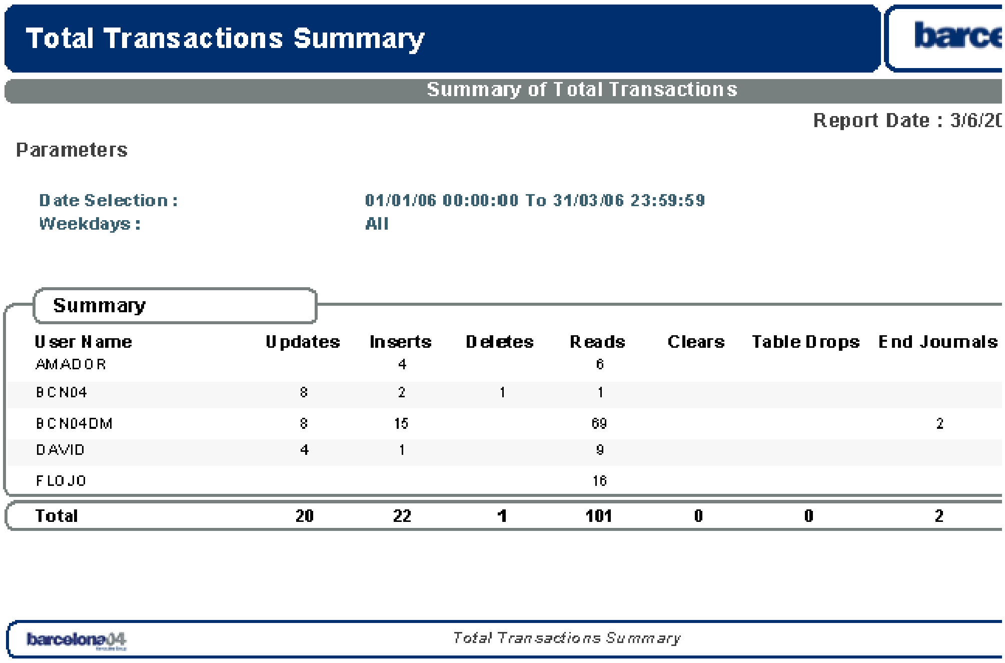
The last page shows a summary of number of changes per operation type, both in table form and as a bar chart.

10.1.4 Total Transactions Summary

The Total Transactions Summary report gives you high level overview of transactions performed on your system data.

It offers a complete list of the subtotals, and totals for each group you choose of every kind of operation. It is very useful for seeing which table has more traffic, which one is useless, and which ones are having more traffic than the expected. It also draws five pie charts to help you finding the most active files. Also to know which user is the most active, which server has more traffic, and other useful information.

In addition you can use it to show statistics of transactions executed on a database by username, File Library/Name, or Program Library/Name.



Total Transactions Summary

barce

Summary of Total Transactions

Report Date : 3/6/20

Parameters

Date Selection : 01/01/06 00:00:00 To 31/03/06 23:59:59
Weekdays : All

Summary

User Name	Updates	Inserts	Deletes	Reads	Clears	Table Drops	End Journals
AMADOR		4		6			
BCN04	8	2	1	1			
BCN04DM	8	15		69			2
DAVID	4	1		9			
FLOJO				16			
Total	**20**	**22**	**1**	**101**	**0**	**0**	**2**

Total Transactions Summary

Figure 71 – Excerpt from Total Transactions Summary

10.1.5 Creating Customized Sub Reports

You can also create sub reports, which are a variation of the existing reports. For instance, you may create a report called “File changes made this month by `QSECOFR`”, by selecting **This Month** as the report date and **QSECOFR** as the Real User, and saving the parameter file with the name you want.

It is easy to customize the reports. For example, you can change the Tango/04 Computing Group logo to your company’s logo, and perform other customizations. For more information on Reports, see the *VISUAL message Center Reports User Guide*.

10.2 Available Report Export Formats

Reports includes a wide range of export formats for your convenience. The following list covers up to the Reports 8.0.

- PDF: a portable file, which is useful as an individual document
- XLS: a spreadsheet, which could be used as input for another report or for calculation of other business data.

- DOC: a Word document, which is useful when you want to include the results in a larger business report written in Word.
- XML: an Extensible Markup Language file.
- CSV: A comma-separated values file is used for the digital storage of data structured in a table of lists form. Useful for moving tabular data between two different computer programs, for example between a database program and a spreadsheet program.
- XLS (Data Only)

10.2.1 Available Report Destinations

You can send the generated report to several destinations. The most common is disk file (save the report to disk), which is the default, but the following destinations are also available:

- Application
- Disk File
- Lotus Domino Database
- Microsoft Mail (MAPI)

For instance, if you want to send a report by email, use the Microsoft Mail destination. You will be prompted for the recipient’s email address. This option needs a MAPI client installed on your desktop.

Chapter 11
Monitoring Data Monitor for iSeries with VISUAL Message Center

There is no need to use any monitoring software with Data Monitor for iSeries. However, if you are a VISUAL Message Center customer, you can take advantage of the advanced monitoring, automation, visualization, and notification features of the VISUAL Message Center SmartConsole to maximize the return on Data Monitor for iSeries. This chapter contains some ideas for getting the most out of both products.

VISUAL Message Center is also a great addition to Data Monitor for iSeries as it provides IT staff with a single point of control, allowing you to access all the information you need from different systems at once, without the need to sign on to each machine.

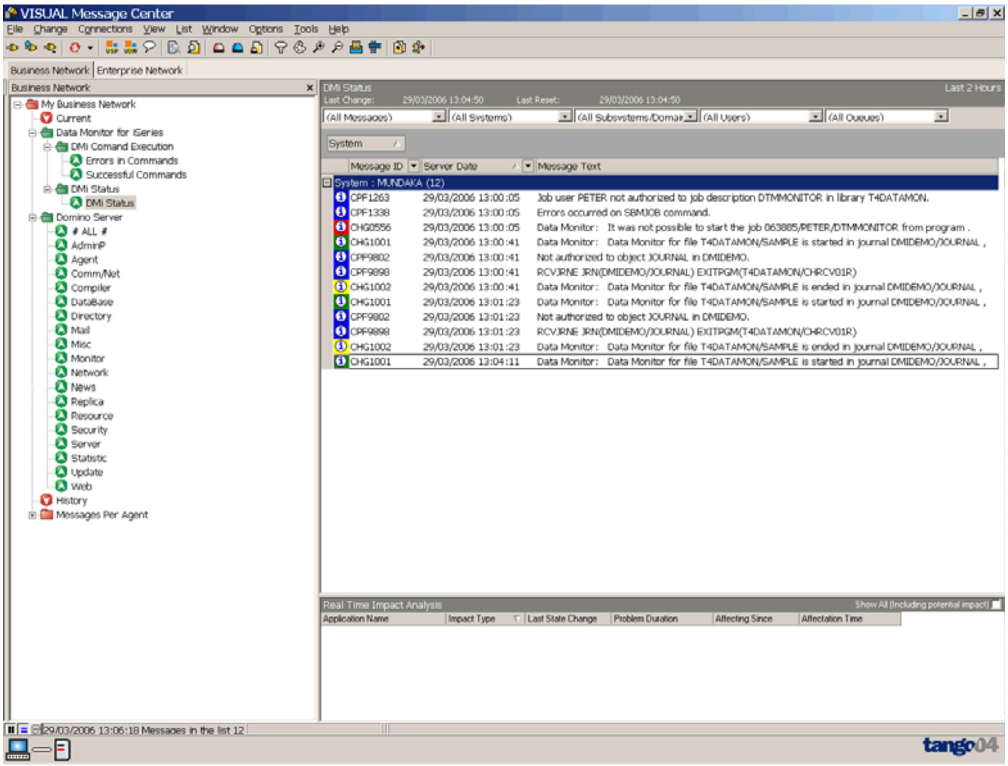
11.1 Monitoring Data Monitor Message Queues

Data Monitor for iSeries stores log messages in several message queues making it easy to integrate Data monitor for iSeries with VISUAL Message Center. Simply use the VISUAL Message Center iSeries Server agent to monitor batch message queues in your system. For details *see section 11.5 - Integrating Data Monitor with VISUAL Message Center on page 89*.

11.2 Creating Data Monitor Business Views

You can create a Business View for Data Monitor for iSeries messages.

© 2013 Tango/04 Computing Group



Last Change: 29/03/2006 13:04:50 Last Reset: 29/03/2006 13:04:50

System : MUNDAKA (12)

Message ID	Server Date	Message Text
CPF1263	29/03/2006 13:00:05	Job user PETER not authorized to job description DTMMONITOR in library T4DATAMON.
CPF1338	29/03/2006 13:00:05	Errors occurred on SBMJOB command.
CHG0556	29/03/2006 13:00:05	Data Monitor: It was not possible to start the job 063885/PETER/DTMMONITOR from program .
CHG1001	29/03/2006 13:00:41	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is started in journal DMIDEMO/JOURNAL ,
CPF9802	29/03/2006 13:00:41	Not authorized to object JOURNAL in DMIDEMO.
CPF9898	29/03/2006 13:00:41	RCVJRNE JRN(DMIDEMO/JOURNAL) EXITPGM(T4DATAMON/CHRCV01R)
CHG1002	29/03/2006 13:00:41	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is ended in journal DMIDEMO/JOURNAL ,
CHG1001	29/03/2006 13:01:23	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is started in journal DMIDEMO/JOURNAL ,
CPF9802	29/03/2006 13:01:23	Not authorized to object JOURNAL in DMIDEMO.
CPF9898	29/03/2006 13:01:23	RCVJRNE JRN(DMIDEMO/JOURNAL) EXITPGM(T4DATAMON/CHRCV01R)
CHG1002	29/03/2006 13:01:23	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is ended in journal DMIDEMO/JOURNAL ,
CHG1001	29/03/2006 13:04:11	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is started in journal DMIDEMO/JOURNAL ,

29/03/2006 13:06:18 Messages in the list 12

Figure 72 – VISUAL Message Center SmartConsole, with an open Event List where Data Monitor for iSeries is grouped by system and message ID. The icon changes from green to red automatically if severe messages are received. Automated actions can be defined easily.

11.3 Receiving and Processing Important Data Monitor for iSeries Events

You can set rules to process and be alerted of important Data Monitor for iSeries Events. For example you can be alerted of:

- Starting and ending events
- Any error in processing transactions
- Unauthorized attempts to change Data Monitor for iSeries configuration (requires iSeries Security Agent)
- `T4DATAMON` subsystem and Data Monitor job’s operating problems
- Result of commands executed for Data Monitor using ALEV expressions

Alerts can be escalated to different groups of people according to the day of the week, time of day, severity, message type and more.

11.4 Creating Enterprise Views to Control Data Monitor for iSeries

VISUAL Message Center SmartConsole excels at showing operational information rapidly, using customizable dashboards called Enterprise Views. You can see all the relevant information about your system at a glance, including hardware and software operating health, security, performance, service levels, product and application related status. See the figure below as an example.

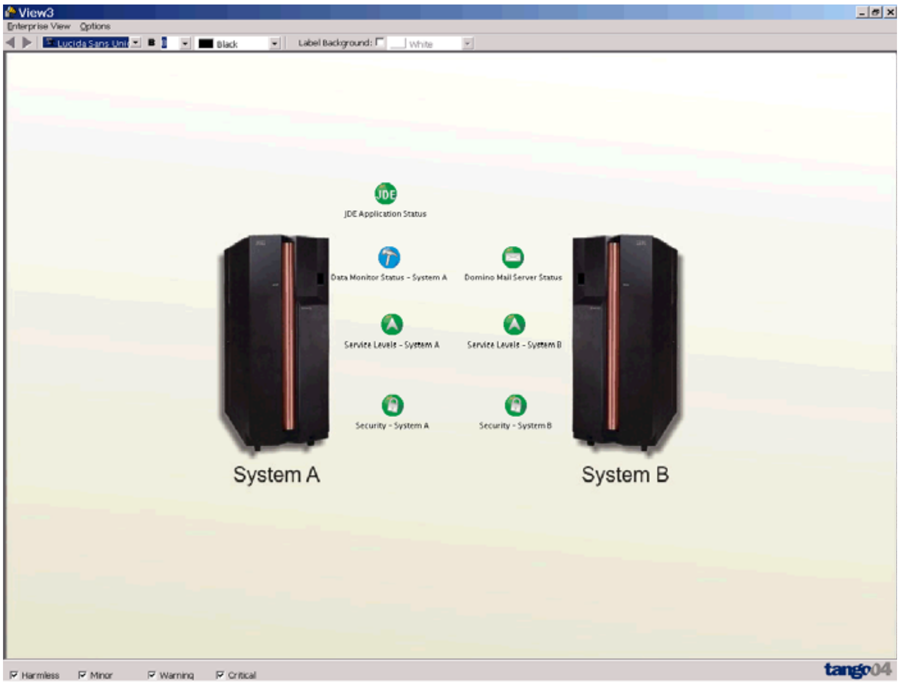


Figure 73 – Enterprise View

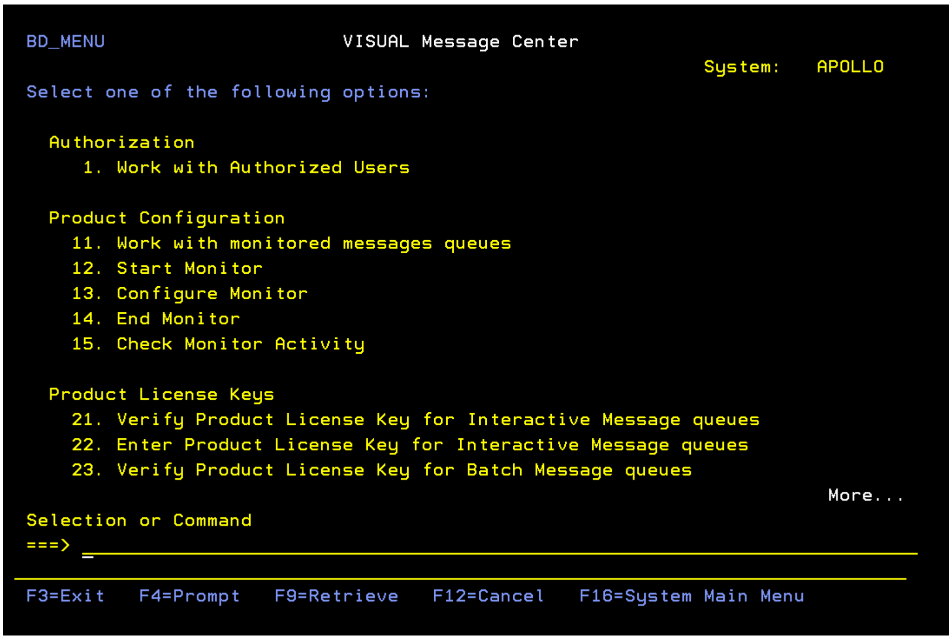
11.5 Integrating Data Monitor with VISUAL Message Center

11.5.1 iSeries-Side Configuration

Data Monitor uses four message queues in the `T4DATAMON` library:

- `T4DATAMON`: all messages for Data Monitor for sensitive files (Starting and ending events, any error in processing transactions, unauthorized attempts to change Data Monitor for iSeries configuration – requires iSeries Security Agent – and `T4DATAMON` subsystem and Data Monitor job's operating problems)
- `T4DATAUSR`: all messages for Data Monitor for sensitive users (Starting and ending events, any error in processing transactions, unauthorized attempts to change Data Monitor for iSeries configuration – requires iSeries Security Agent – and `T4DATAMON` subsystem and Data Monitor job's operating problems)
- `GENJRNDTA`: all messages for Data Monitor Express (`GENJRNDTA` command) (Starting and ending events, any error in processing transactions, unauthorized attempts to change Data Monitor for iSeries configuration – requires iSeries Security Agent – and `T4DATAMON` subsystem and Data Monitor job's operating problems)
- `DTMCMDS`: Result of commands executed for Data Monitor using ALEV expressions

If you want these messages to be monitored using VISUAL Message Center technology, you must add these message queues to the list of monitored message queues, using the `B_DETECTOR/WRKBDMSGQ` command, or using option **11** of the `B_DETECTOR/BD_MENU` menu.

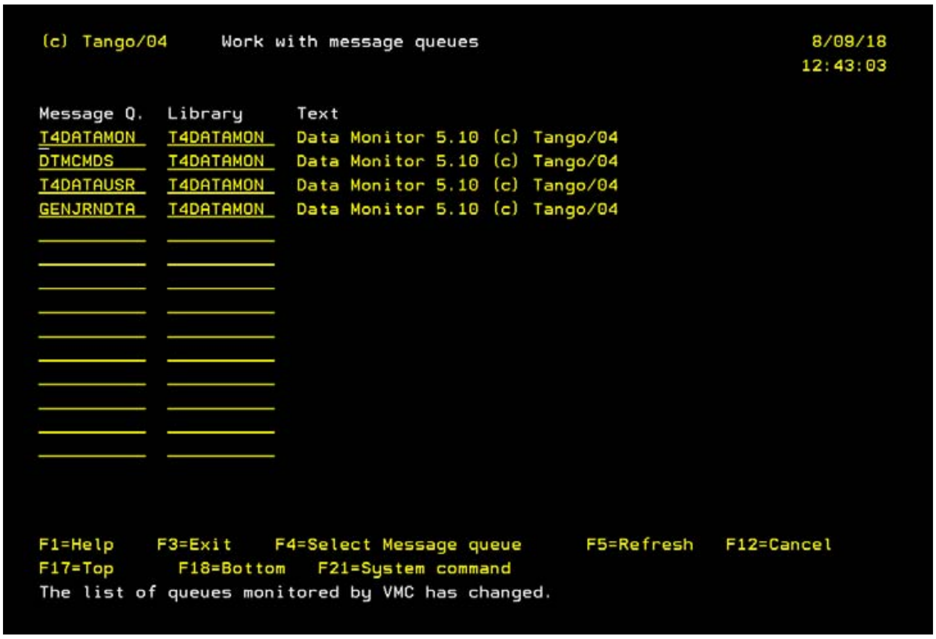


```
BD_MENU                    VISUAL Message Center
                                                       System:    APOLLO
Select one of the following options:

  Authorization
     1. Work with Authorized Users

  Product Configuration
    11. Work with monitored messages queues
    12. Start Monitor
    13. Configure Monitor
    14. End Monitor
    15. Check Monitor Activity

  Product License Keys
    21. Verify Product License Key for Interactive Message queues
    22. Enter Product License Key for Interactive Message queues
    23. Verify Product License Key for Batch Message queues
                                                                  More...
Selection or Command
===> _

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F16=System Main Menu
```

Figure 74 – VISUAL Message Center main menu

The list of messages monitored by VISUAL Message Center’s batch monitor appears. Add the following four message queues here:

- `T4DATAMON/T4DATAMON`
- `T4DATAMON/DTMCMDS`
- `T4DATAMON/T4DATAUSR`
- `T4DATAMON/GENJRNDTA`



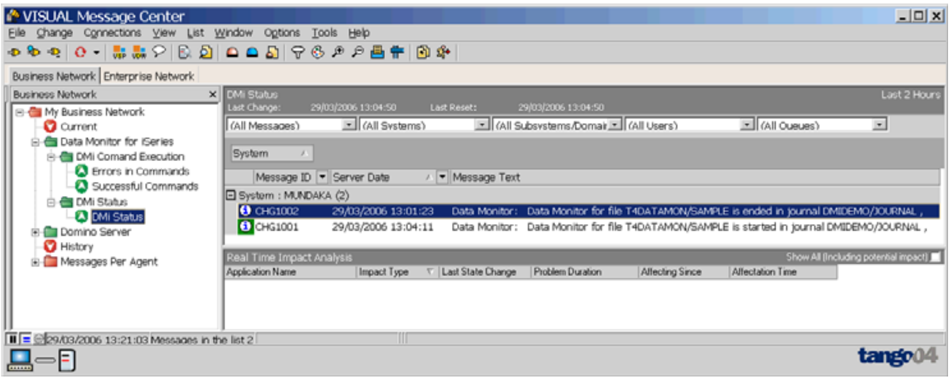
```
 (c) Tango/04     Work with message queues                          8/09/18
                                                                   12:43:03

 Message Q.   Library     Text
 T4DATAMON    T4DATAMON   Data Monitor 5.10 (c) Tango/04
 DTMCMDS      T4DATAMON   Data Monitor 5.10 (c) Tango/04
 T4DATAUSR    T4DATAMON   Data Monitor 5.10 (c) Tango/04
 GENJRNDTA    T4DATAMON   Data Monitor 5.10 (c) Tango/04

 F1=Help    F3=Exit    F4=Select Message queue      F5=Refresh    F12=Cancel
 F17=Top      F18=Bottom    F21=System command
 The list of queues monitored by VMC has changed.
```

Figure 75 – Work with message queues

The VISUAL Message Center batch monitor must be active for these message queues to be monitored. Check the batch monitor is active using the command `B_DETECTOR/CHKBDMON`. If necessary, start the batch monitor with the command `B_DETECTOR/STRBDMON`.

11.5.2 Console-Side Configuration

You can create folders and Business Views in the VISUAL Message Center SmartConsole to help you manage the messages coming from the Data Monitor for iSeries.



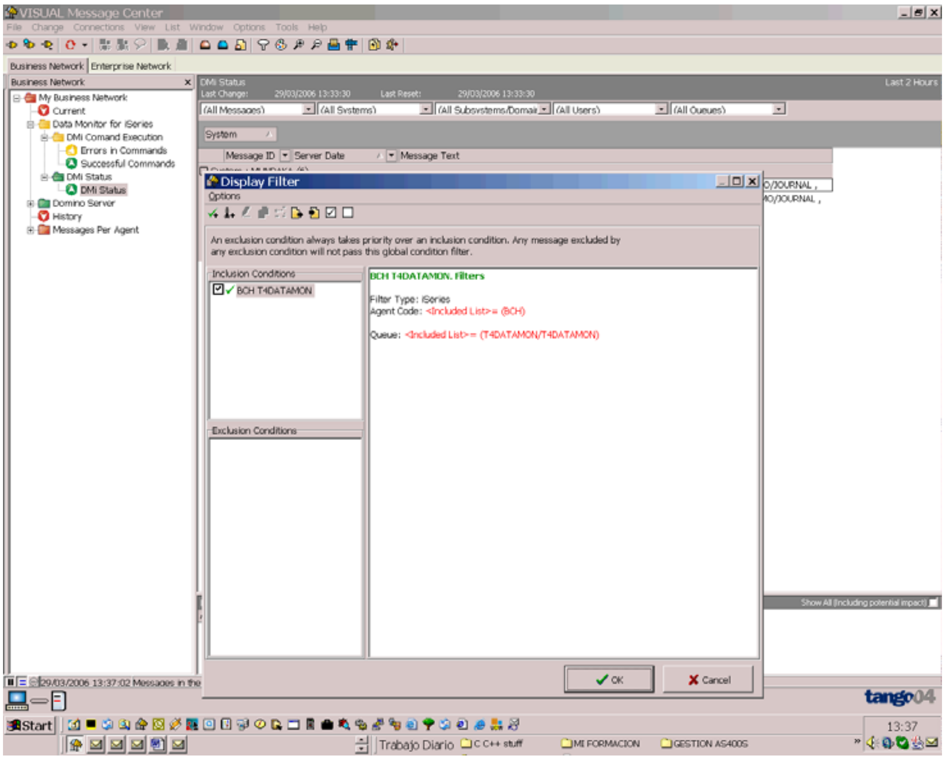
Last Change: 29/03/2006 13:04:50 Last Reset: 29/03/2006 13:04:50

Message ID	Server Date	Message Text
System : MUNDAKA (2)		
CHG1002	29/03/2006 13:01:23	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is ended in journal DMIDEMO/JOURNAL ,
CHG1001	29/03/2006 13:04:11	Data Monitor: Data Monitor for file T4DATAMON/SAMPLE is started in journal DMIDEMO/JOURNAL ,

29/03/2006 13:21:03 Messages in the list 2

Figure 76 – VISUAL Message Center SmartConsole

For example, you can create a Business View for each message queue:



Last Change: 29/03/2006 13:33:30 Last Reset: 29/03/2006 13:33:30

An exclusion condition always takes priority over an inclusion condition. Any message excluded by any exclusion condition will not pass this global condition filter.

☑ ✓ BCH T4DATAMON

BCH T4DATAMON. Filters

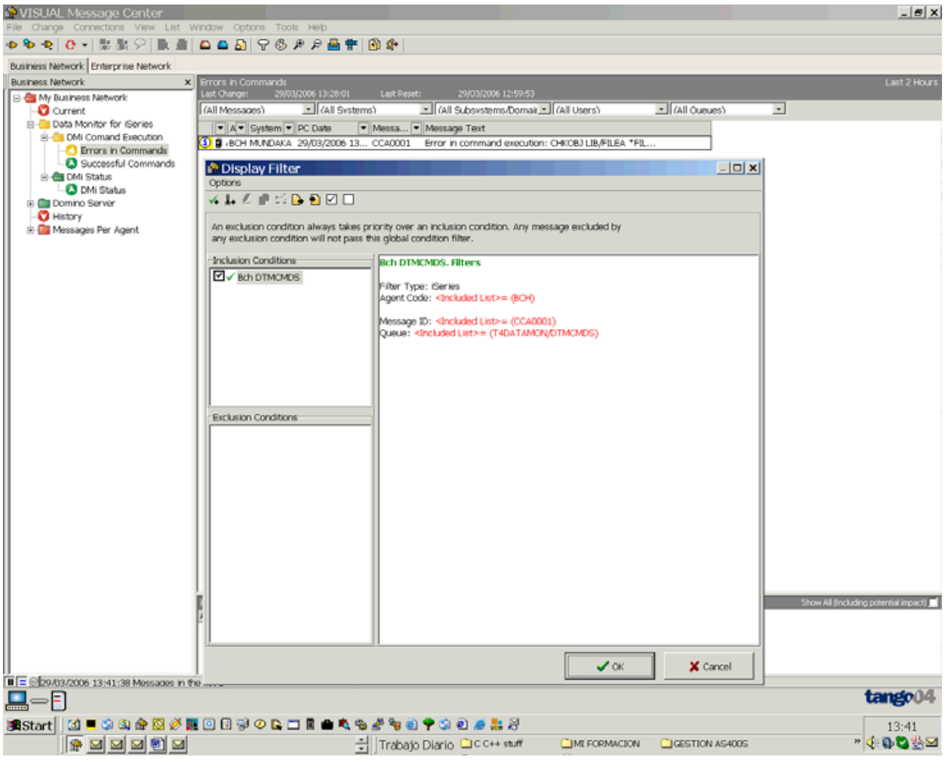
Filter Type: iSeries
Agent Code: <Included List>= (BCH)

Queue: <Included List>= (T4DATAMON/T4DATAMON)

29/03/2006 13:37:02 Messages in the

Figure 77 – Business Views in the SmartConsole

You can also create Business Views for commands executed from ALEV Preprocess expressions as shown in the next two figures.



Errors in Commands — Last 2 Hours
Last Change: 29/03/2006 13:28:01 Last Reset: 29/03/2006 12:59:53

(All Messages) (All Systems) (All Subsystems/Domair (All Users) (All Queues)

A	System	PC Date	Messa...	Message Text
	BCH MUNDAKA	29/03/2006 13...	CCA0001	Error in command execution: CHKOBJ LIB/FILEA *FIL...

An exclusion condition always takes priority over an inclusion condition. Any message excluded by any exclusion condition will not pass this global condition filter.

☑ ✓ Bch DTMCMDS

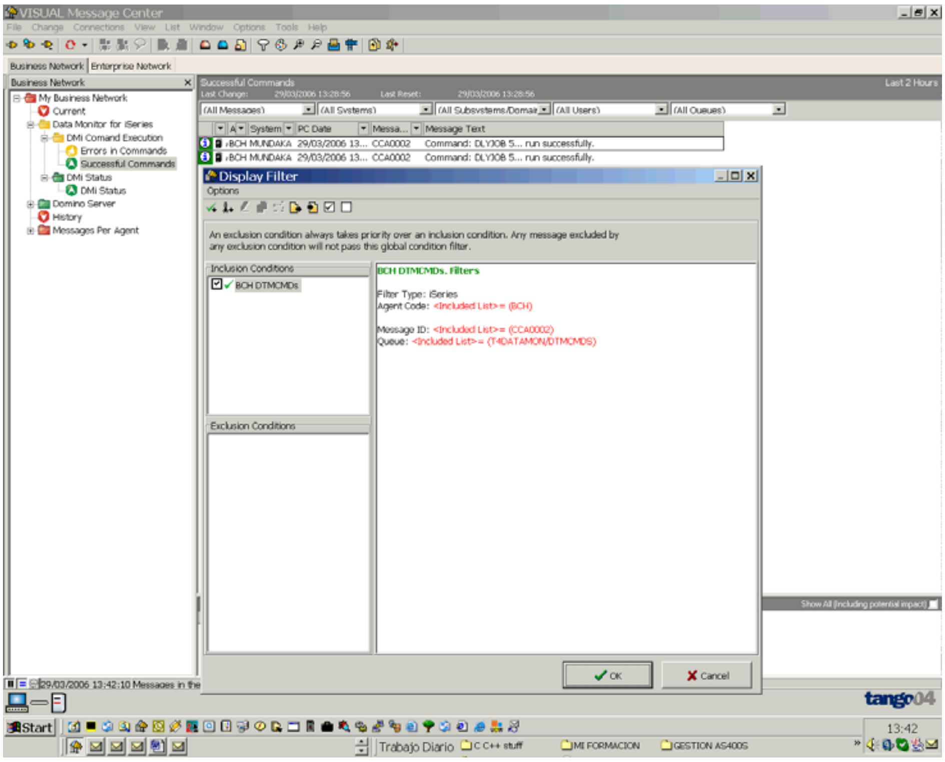
Bch DTMCMDS. Filters

Filter Type: iSeries
Agent Code: <Included List>= (BCH)

Message ID: <Included List>= (CCA0001)
Queue: <Included List>= (T4DATAMON/DTMCMDS)

29/03/2006 13:41:38 Messages in the ...

Figure 78 – Example 1 of Business View for commands executed from ALEV Preprocess expressions in the SmartConsole



Successful Commands — Last 2 Hours
Last Change: 29/03/2006 13:28:56 Last Reset: 29/03/2006 13:28:56

(All Messages) (All Systems) (All Subsystems/Domair (All Users) (All Queues)

A	System	PC Date	Messa...	Message Text
	BCH MUNDAKA	29/03/2006 13...	CCA0002	Command: DLYJOB 5... run successfully.
	BCH MUNDAKA	29/03/2006 13...	CCA0002	Command: DLYJOB 5... run successfully.

An exclusion condition always takes priority over an inclusion condition. Any message excluded by any exclusion condition will not pass this global condition filter.

☑ ✓ BCH DTMCMDs

BCH DTMCMDs. Filters

Filter Type: iSeries
Agent Code: <Included List>= (BCH)

Message ID: <Included List>= (CCA0002)
Queue: <Included List>= (T4DATAMON/DTMCMDS)

29/03/2006 13:42:10 Messages in the

Figure 79 – Example 2 of Business View for commands executed from ALEV Preprocess expressions in the SmartConsole

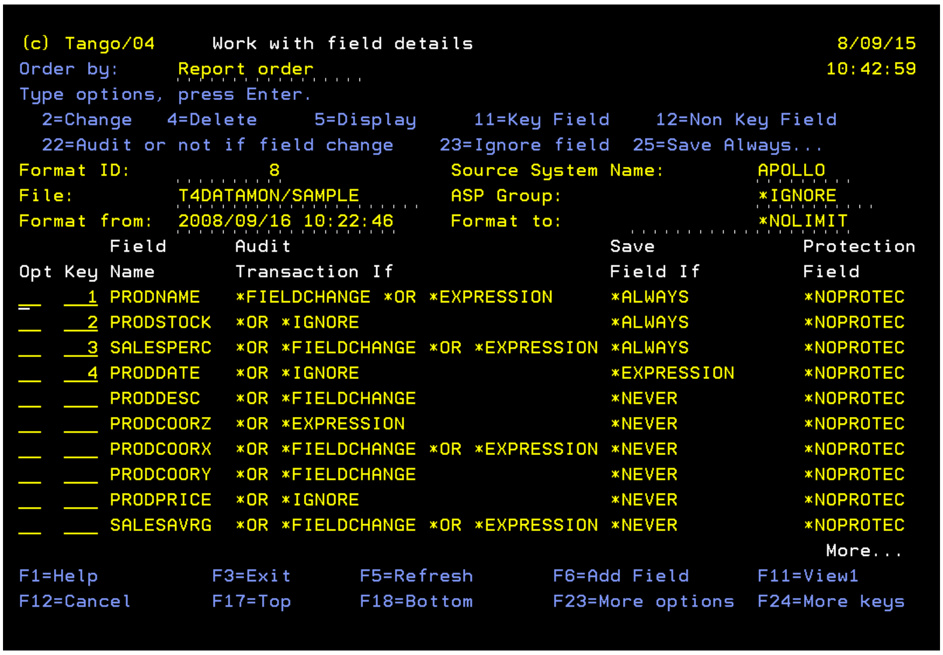
Appendix A

Recommendations for Protecting High Level Data

Current data protections laws and new proposed European Data Protection Directives stipulate specific control of the information that companies store in their databases. It is therefore vital to have the ability to track confidential data and make sure it is protected wherever it may be stored.

Data Monitor for iSeries has the power to apply flexible protection to data depending on its level of sensitivity. How to set these controls is explained in *section 3.5.3 - View 3 - Auditing and Protection Details* on *page 27*.

The auditing and protection details for each field can be edited to ensure the correct level of protection.



```
(c) Tango/04     Work with field details                               8/09/15
Order by:      Report order                                           10:42:59
Type options, press Enter.
  2=Change   4=Delete     5=Display     11=Key Field    12=Non Key Field
  22=Audit or not if field change     23=Ignore field  25=Save Always...
Format ID:            8                 Source System Name:        APOLLO
File:          T4DATAMON/SAMPLE         ASP Group:                 *IGNORE
Format from:   2008/09/16 10:22:46      Format to:                 *NOLIMIT
        Field      Audit                            Save             Protection
Opt Key Name       Transaction If                   Field If         Field
__   __1 PRODNAME   *FIELDCHANGE *OR *EXPRESSION     *ALWAYS          *NOPROTEC
__   __2 PRODSTOCK  *OR *IGNORE                      *ALWAYS          *NOPROTEC
__   __3 SALESPERC  *OR *FIELDCHANGE *OR *EXPRESSION *ALWAYS          *NOPROTEC
__   __4 PRODDATE   *OR *IGNORE                      *EXPRESSION      *NOPROTEC
__   ___ PRODDESC   *OR *FIELDCHANGE                 *NEVER           *NOPROTEC
__   ___ PRODCOORZ  *OR *EXPRESSION                  *NEVER           *NOPROTEC
__   ___ PRODCOORX  *OR *FIELDCHANGE *OR *EXPRESSION *NEVER           *NOPROTEC
__   ___ PRODCOORY  *OR *FIELDCHANGE                 *NEVER           *NOPROTEC
__   ___ PRODPRICE  *OR *IGNORE                      *NEVER           *NOPROTEC
__   ___ SALESAVRG  *OR *FIELDCHANGE *OR *EXPRESSION *NEVER           *NOPROTEC
                                                                        More...
F1=Help          F3=Exit       F5=Refresh       F6=Add Field      F11=View1
F12=Cancel       F17=Top       F18=Bottom       F23=More options  F24=More keys
```

Figure 80 – Work with field details

There are three levels of protection:

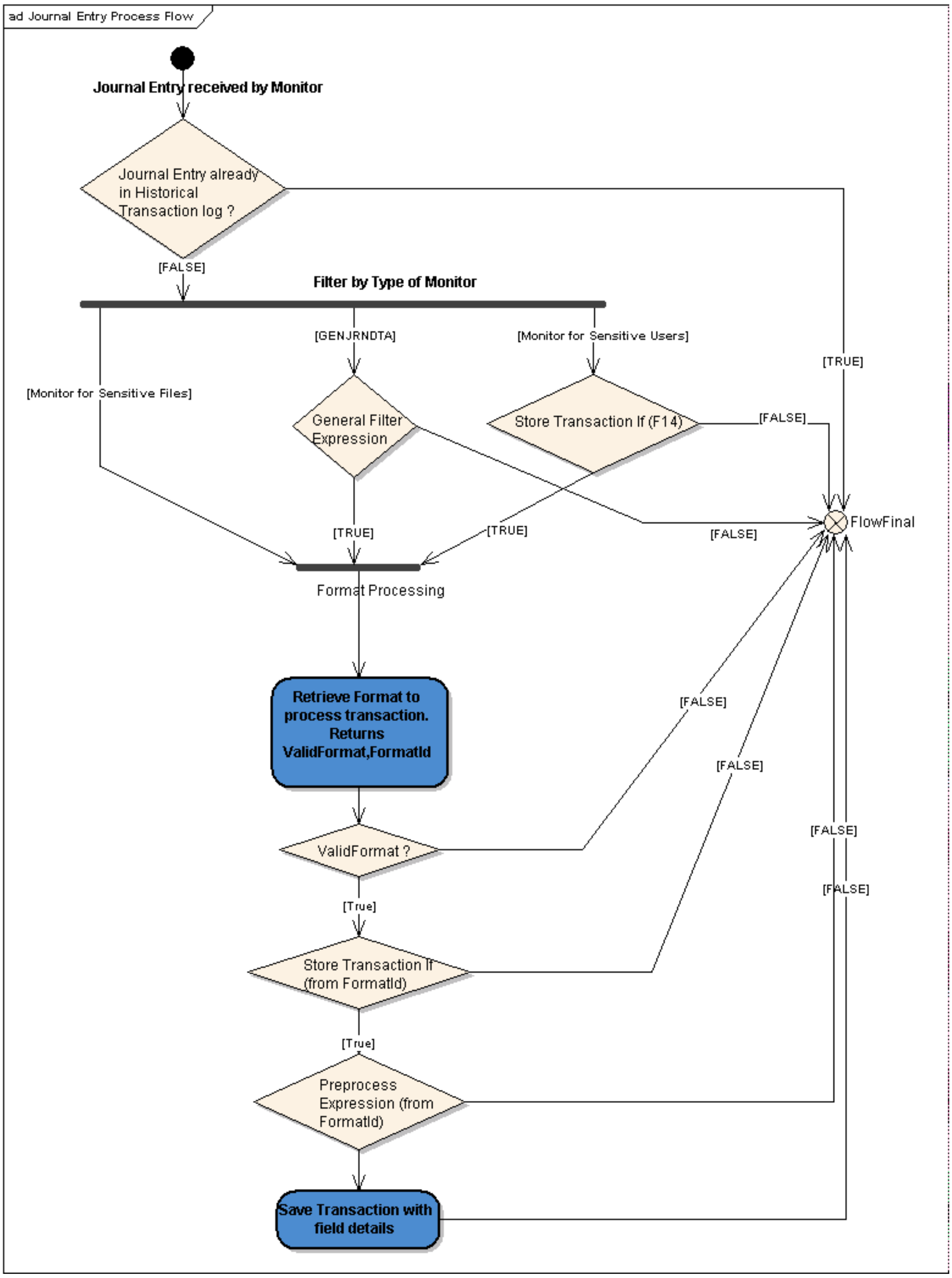
- `*NOPROTEC`: the value is stored and displayed in Data Monitor reports
- `*SOFT`: the value is stored, but not displayed in reports. Instead of the value the report will show *Restricted*
- `*HARD`: the value is neither stored nor displayed in the report. The word `*PROTECTED*` appears in the report instead of the value.

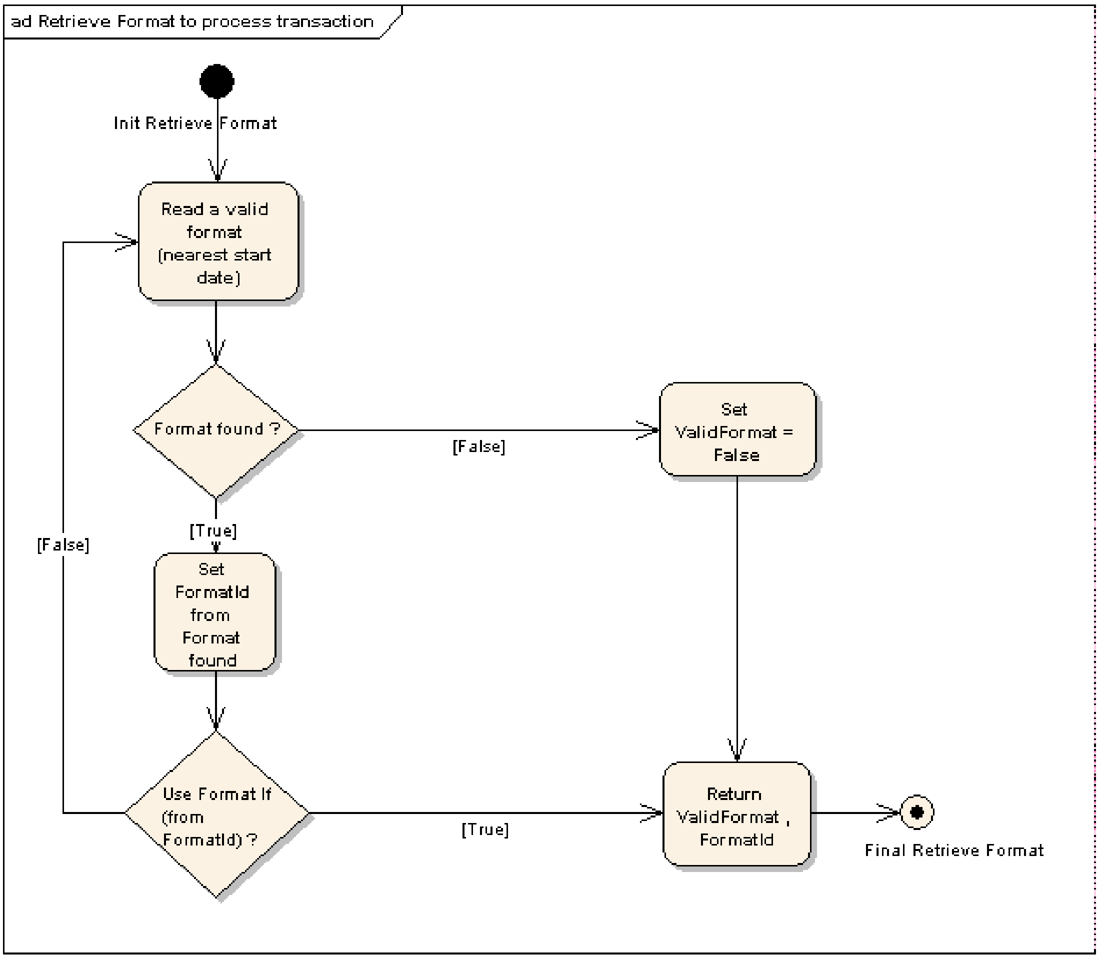
Being able to apply *Hard* protection is critical when you store High Level sensitive data in your databases, such as medical data, details of sexual orientation, religious beliefs, or political affiliations.

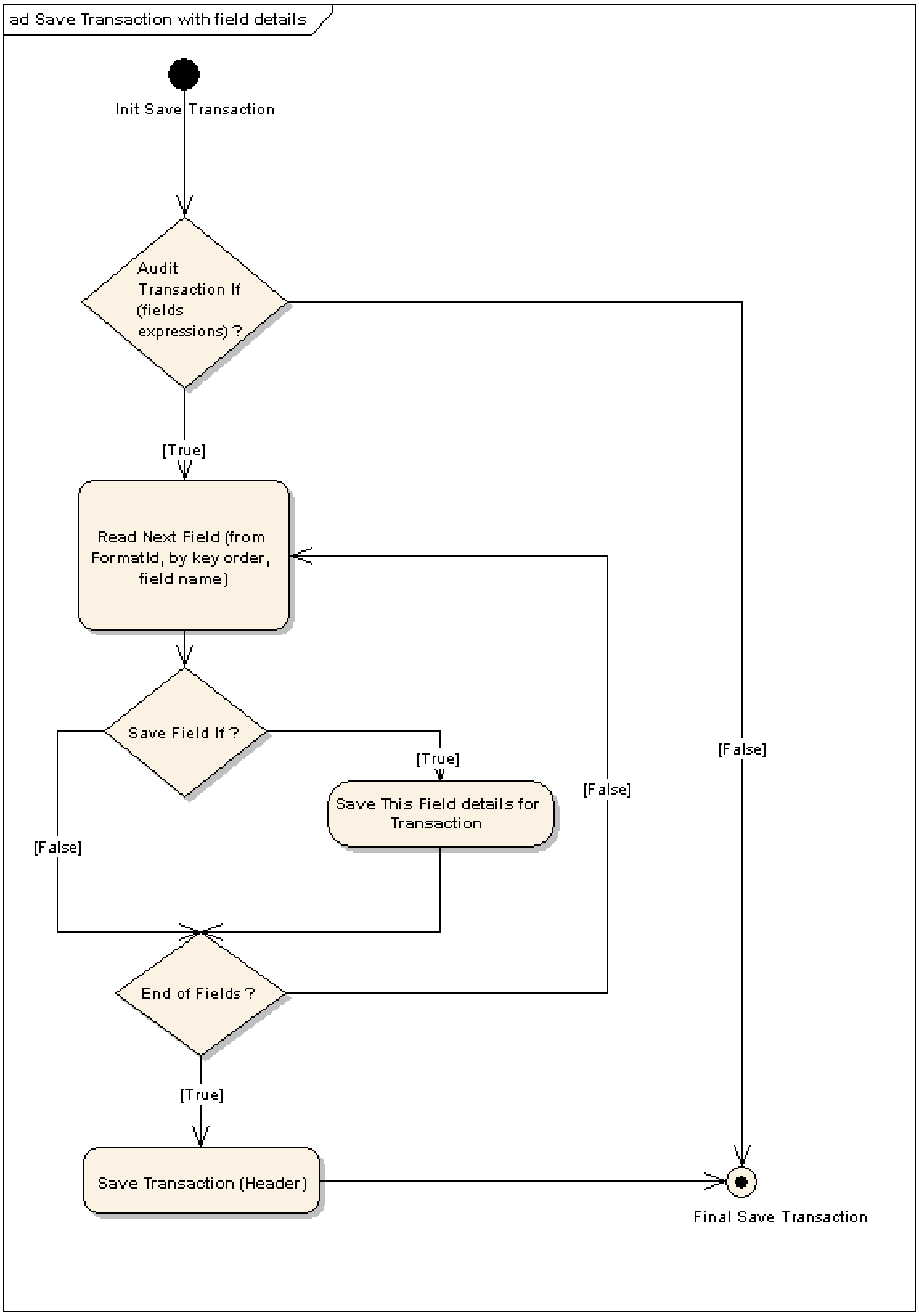
Hard protection ensures that data will not be copied or transferred to other databases during an audit for example, and also that the information will not appear in reports generated from your systems.

Appendix B

Journal Entry Process Flow


ad Journal Entry Process Flow
Journal Entry received by Monitor
Journal Entry already in Historical Transaction log ?
[FALSE]
[TRUE]
Filter by Type of Monitor
[Monitor for Sensitive Files]
[GENJRNDTA]
[Monitor for Sensitive Users]
General Filter Expression
Store Transaction If (F14)
[FALSE]
[TRUE]
[TRUE]
[FALSE]
FlowFinal
Format Processing
Retrieve Format to process transaction. Returns ValidFormat,FormatId
ValidFormat ?
[FALSE]
[True]
Store Transaction If (from FormatId)
[FALSE]
[True]
Preprocess Expression (from FormatId)
[FALSE]
Save Transaction with field details
[FALSE]

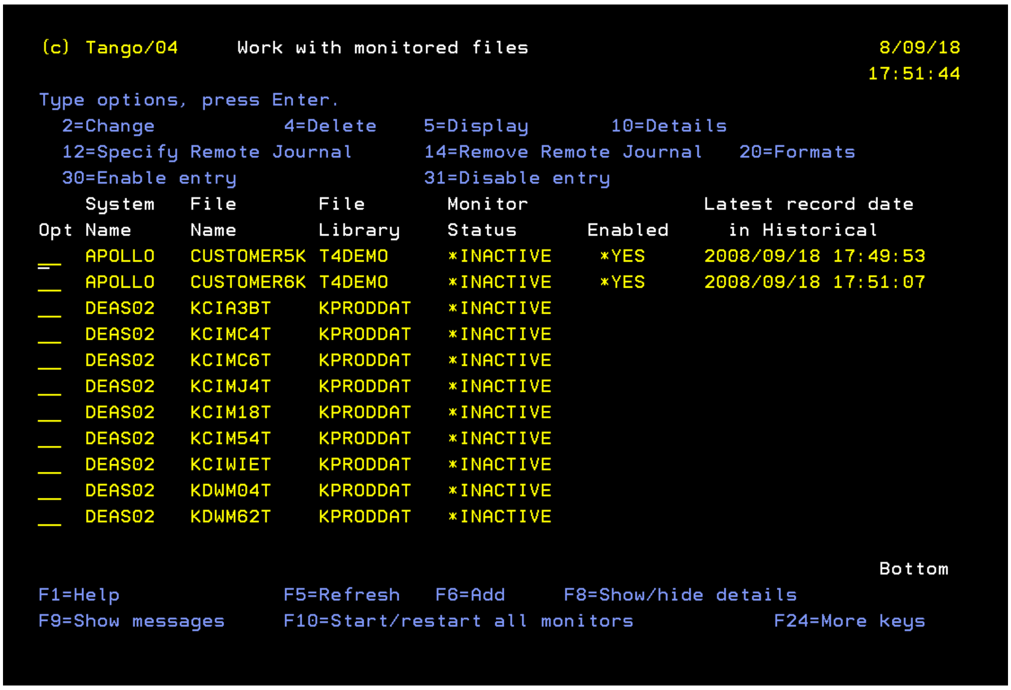

ad Retrieve Format to process transaction
Init Retrieve Format
Read a valid format (nearest start date)
Format found ?
[False]
Set ValidFormat = False
[True]
[False]
Set FormatId from Format found
Use Format If (from FormatId) ?
[True]
Return ValidFormat , FormatId
Final Retrieve Format


ad Save Transaction with field details
Init Save Transaction
Audit Transaction If (fields expressions) ?
[True]
[False]
Read Next Field (from FormatId, by key order, field name)
Save Field If ?
[True]
[False]
Save This Field details for Transaction
[False]
End of Fields ?
[True]
Save Transaction (Header)
Final Save Transaction

Appendix C
How to Re-Process Entries

What happens if your data is processed incorrectly, for example, due to a change of the file format that was not configured properly in Data Monitor at the time the entries were processed? Data Monitor offers the flexibility to redo the processing of Data Monitor entries, as long as the entries in question are still stored in the journal receiver. Here is an explanation of steps to take to reprocess entries for Data Monitor for Sensitive Files.

First end the monitor. You can end the monitor by pressing F11 from the Work with monitored files screen. Indicate if you want to remove the trigger for Read transactions.



```
(c) Tango/04     Work with monitored files                            8/09/18
                                                                     17:51:44
Type options, press Enter.
  2=Change           4=Delete     5=Display       10=Details
  12=Specify Remote Journal       14=Remove Remote Journal    20=Formats
  30=Enable entry                 31=Disable entry
    System   File       File       Monitor                  Latest record date
Opt Name     Name       Library    Status      Enabled        in Historical
__  APOLLO   CUSTOMER5K T4DEMO     *INACTIVE    *YES       2008/09/18 17:49:53
__  APOLLO   CUSTOMER6K T4DEMO     *INACTIVE    *YES       2008/09/18 17:51:07
__  DEAS02   KCIA3BT    KPRODDAT   *INACTIVE
__  DEAS02   KCIMC4T    KPRODDAT   *INACTIVE
__  DEAS02   KCIMC6T    KPRODDAT   *INACTIVE
__  DEAS02   KCIMJ4T    KPRODDAT   *INACTIVE
__  DEAS02   KCIM18T    KPRODDAT   *INACTIVE
__  DEAS02   KCIM54T    KPRODDAT   *INACTIVE
__  DEAS02   KCIWIET    KPRODDAT   *INACTIVE
__  DEAS02   KDWM04T    KPRODDAT   *INACTIVE
__  DEAS02   KDWM62T    KPRODDAT   *INACTIVE

                                                                      Bottom
F1=Help              F5=Refresh    F6=Add     F8=Show/hide details
F9=Show messages     F10=Start/restart all monitors            F24=More keys
```

Figure 81 – Work with monitored files screen

Another way to end the monitors is with option **12** in the main menu of the Data Monitor.

Next you must identify the entries you want to reprocess.

These entries will be deleted from the Data Monitor files based on a date range.

From the main menu of Data Monitor use option **22** – Delete historical records. Select the date range for the entries you want to re-process, set audit type to `*ALL`, and enter the file from which you want to reprocess transactions.

Before removing any entries from Data Monitor please make sure that the entries are still stored in the system journal receivers or can be restored otherwise. If this is not the case, these entries will be lost.

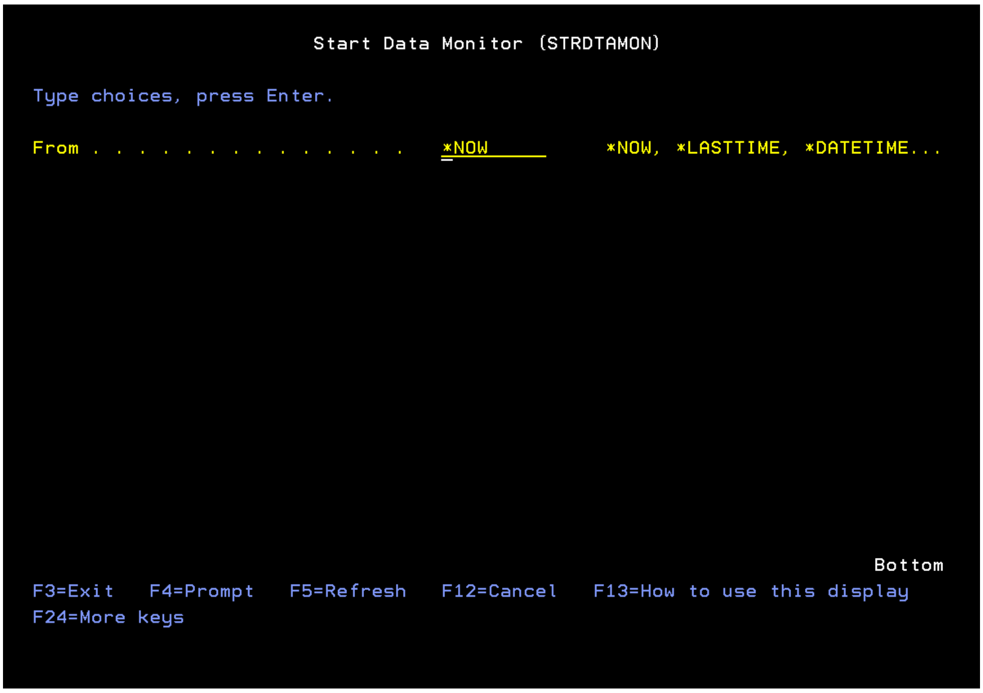
Now add the required formats to the Data Monitor Configuration to reprocess the entries as desired.

Note
You can avoid manually adding the required formats if you decide that you can rely on automatic refresh for formats.

Next enter the date from which you wish to reprocess the entries.

Note
If you want to rely on automatic refresh for formats, then *end journal physical file* and *start journal physical file* events should be included in the date range to reprocess. *See section 7.4 - Automatic Refresh for Formats on page 67* for more details.

You can do this using option F10 start/restart monitors from the Work with monitored files menu. Once you press F10 a screen will appear allowing you to select the date.



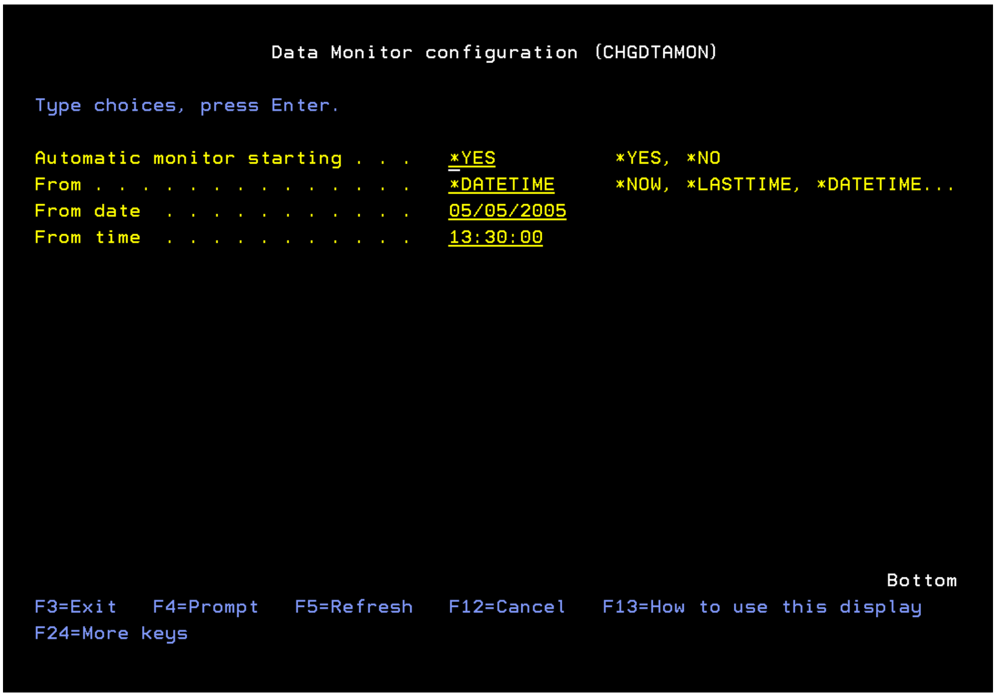
```
                         Start Data Monitor (STRDTAMON)

 Type choices, press Enter.

 From . . . . . . . . . . . . . .   *NOW          *NOW, *LASTTIME, *DATETIME...












                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 82 – Enter the date / time to start the monitor

Alternatively you can set the time in the main menu of Data Monitor using option **13** – Automatic start settings. You can also access the automatic start settings with the following command:

```
T4DATAMON/CHGDTAMON
```

You can select `*NOW`, `*LASTTIME` or `*DATETIME`. If you select `*DATETIME` you will be able to enter the date and time from which you want to start processing the entries (*see section 3.6 - Starting/Restarting Monitors on page 30* for details).



```
                   Data Monitor configuration (CHGDTAMON)

Type choices, press Enter.

Automatic monitor starting . . .   *YES          *YES, *NO
From . . . . . . . . . . . . . .   *DATETIME     *NOW, *LASTTIME, *DATETIME...
From date  . . . . . . . . . . .   05/05/2005
From time  . . . . . . . . . . .   13:30:00

                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 83 – Automatic start settings

The next time you start the monitor, events will be processed using the formats you configured in these steps and from the date and time you specified here. Finally, start the monitors.

From the Data Monitor's main menu use option **11** – Start monitoring files for changes. Alternatively, go to the work with monitored files screen and press F10.

Note

You can reprocess entries for Data Monitor for Sensitive Users by following the same steps, simply using the appropriate commands for Data Monitor for Sensitive Users.

Appendix D
How to Reduce Occupied Disk Space

By default every transaction stores values for all fields in the file format, which means that the disk space that Data Monitor internal files occupy can grow dramatically (*see also Appendix K: Frequently Asked Questions on page 147*). Fortunately there is an easy way to reduce the number of fields stored and thus the amount of disk space occupied by the Data Monitor.

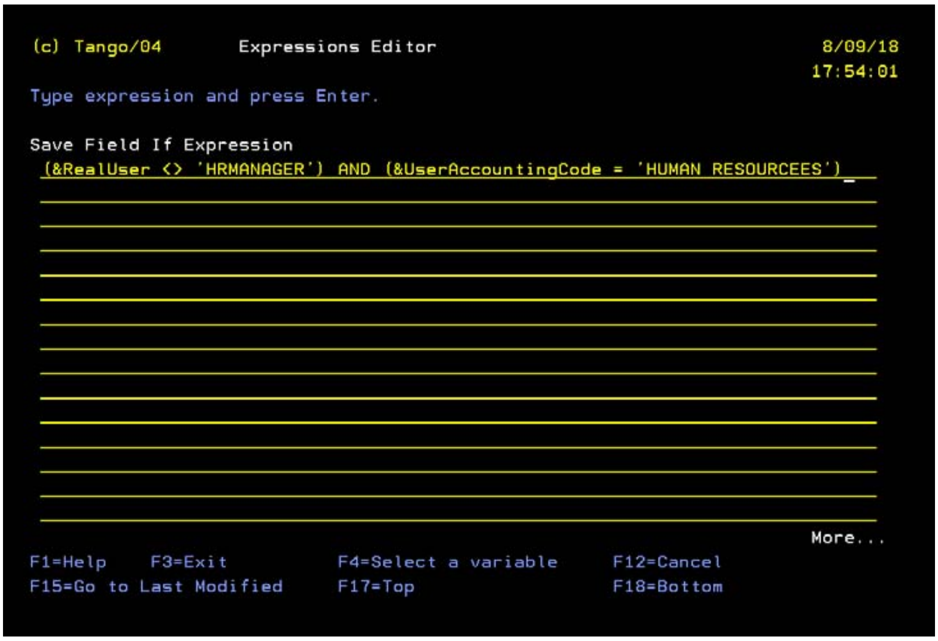
There are several ways to prevent storing unimportant fields:

D.0.1 Option 41 – Do not save field

In Work with field details set Save Field If to NEVER using option **41** – Do not save field. The fields will not be stored and not occupy unnecessary disk space.

D.0.2 Option 40 – Edit save field expression

In Work with field details use option **40** – Edit save field expression to add an expression based on which the field will be stored or not. If the statement is false the field will not be stored thus not unnecessarily occupying disk space.



```
(c) Tango/04        Expressions Editor                                  8/09/18
                                                                       17:54:01
Type expression and press Enter.

Save Field If Expression
 (&RealUser <> 'HRMANAGER') AND (&UserAccountingCode = 'HUMAN RESOURCEES')_

                                                                       More...
F1=Help    F3=Exit             F4=Select a variable       F12=Cancel
F15=Go to Last Modified        F17=Top                    F18=Bottom
```

Figure 84 – Expression Editor

For example: The HR manager is the only person in the HR department who should access the `SALARIES` file. We would like to monitor the `SALARIES` file to make sure no one else accesses the file and, if this were to happen, that the details are recorded. The expression in the figure above states that if the user is not HR Manager and does belong to the Human Resources accounting group, the field should be stored.

D.0.3 Option 30 - Edit audit transaction if

In Work with field details use option **30** – Edit audit transaction if to add an expression based on which the transaction will be stored or not. Only if the expression is true will the transaction be stored.

D.1 Replicating the Data Monitor Database

Another way to reduce disk space is to replicate Data Monitor’s database (Historical Transaction Log) located on the iSeries to another platform such as Oracle or SQL Server.

Leading database engines such as Microsoft SQL Server, Oracle, or IBM UDB DB2 include replication services, which can help you replicate the data in real time, near real time, or in bulk transfers. Additionally, there are some third-party tools that also provide replication services, and your company may be using one. You may also use simple SQL statements to copy the data, periodically sending the changes to the target database and deleting the source data on the iSeries system once copied.

Check with your database administrator for the destination database or refer to the destination database documentation to learn more about how to synchronize the data in the manner most convenient for you.

Tip
When using Data Monitor for iSeries to monitor Users it only makes sense to replicate data if the same users are present on the destination machine. We suggest you review the user list once you have replicated the data.

Once it is replicated you can use Reports, which uses an ODBC connection, to retrieve data from Data Monitor’s replicated database.

Finally after replicating this data you can clean database in the iSeries system.

For details *see Chapter 7 - Cleaning Data Monitor files on page 61*.

Appendix E
Formats with Overlapping Date Ranges

This appendix explains the priority rule for overlapping date ranges.

If, for a certain transaction date, there is more than one format definition that could potentially be used, by default the format with the *Format valid FROMDATE* that is closest to the transaction date will be used.

For example, suppose you have three formats,

- One with `FROMDATE  *NOLIMIT`,
- One with `FROMDATE` 1/1/2004 and
- One with `FROMDATE` 1/1/2005

And we assume the three of them have `*NOLIMIT` as `TODATE`, their effect is as follows:

- Transaction 1, dated December 31, 2003: (A) will be used, since (B) and (C) are still not valid at that date.
- Transaction 2, dated 1/1/2004: (B) will be used, since (B) has the closest `FROMDATE` between (A) and (B), and (C) cannot be used since will not be in effect until 1/1/2005.
- Transaction 3, dated 12/12/2004: (B) will be used, for the same reasons as transaction 2.
- Transaction 4, dated 1/1/2005: (C) will be used, since (C) has the closest `FROMDATE` among all the format definitions.
- Any transaction dated after 1/1/2005 will use the (C) format, for the same reason as above.
- Any transaction dated before 1/1/2004 will use the (A) format since (B) and (C) are not yet valid at these dates.

The `FROMDATE` parameter is always the deciding factor: if two format definitions are valid for a certain date, the `TODATE` parameter has no major effect on determining which format definition will be used. However, the `TODATE` parameter does have an effect in determining whether a format definition is or is not valid for a date range.

If more than one format definition applies to a certain transaction date and the parameter `USEIF` has been defined, the `USEIF` conditional expression is tested for all the applicable format definitions in the order defined in the previous rule until one format definition satisfies the `USEIF` expression. For more information on defining the `USEIF` conditional expression *see section - Expression to use format (option 20) on page 21*.

Appendix F
Replicating iSeries Configuration

To replicate your configuration from one iSeries to another, you must first make sure Data Monitor is installed on your target server. If it is not yet installed on your target server, first install Data Monitor.

Next copy the objects in the table below with your replication tool of choice. You can also duplicate the objects manually (for example with FTP). All of these objects are located in the library `T4DATAMON`.

F.1 Files (*FILE)

Replicate the files relevant to the agent configuration you want to copy from the source machine to the target machine.

Before starting the replication, make sure that all Data Monitor monitors are ended and that you have exited all Data Monitor menus in the target system.

File	Files	Users	Express	Description	How to get there
CHDOF01P	x	x	x	Detail of fields for every format of files monitored by Data Monitor.	`GO T4DATAMON/DTM_CONFIG`, option 1 and then option 20 for any file. Finally select option 10 for any format (only available from v1.20).
CHHOF01P	x	x	x	History of formats. It contains all formats for every file monitored by Data Monitor.	`GO T4DATAMON/DTM_CONFIG`, option 1 and then option 20 for any file in list of files monitored by Data Monitor.
CHITP01P	x			List of files monitored by Data Monitor.	`GO T4DATAMON/DTM_CONFIG`, option 1.
CHODC01P			x	The Data Monitor Express configuration file.	`GO T4DATAMON/DTM_CONFIG`, option 3.

© 2013 Tango/04 Computing Group

File	Files	Users	Express	Description	How to get there
CHOFA01P	x			Remote journals for files monitored by Data Monitor not in current system.	`GO T4DATAMON/DTM_CONFIG`, option 1 and then option 12 for any file in list of files monitored by Data Monitor (only available from v1.20).
CHUGC01P		x		User General Configuration. It contains the general configuration for all the users to monitor, including the transaction to store if expression (F14) and the monitoring options (F15).	`GO T4DATAMON/DTM_CONFIG`, option 2
CHUSR01P		x		List of users monitored by Data Monitor.	`GO T4DATAMON/DTM_CONFIG`, option 2.

After replicating these files carry out the following procedures on the target system. In our examples we assume the source system name is `SOURCE` and the target system name is `TARGET` (this applies only if `SOURCE` and `TARGET` are different):

- Run SQL statement:

```
UPDATE  T4DATAMON/CHITP01P  SET  ITPSSYST  =  'TARGET'  WHERE  ITPSSYST  =
'SOURCE'
```

- Run Command:

```
CPYF      FROMFILE(T4DATAMON/CHHOF01P)      TOFILE(T4DATAMON/CHHOF01P_2)
MBROPT(*ADD) CRTFILE(*YES)
```

- Run SQL statement:

```
UPDATE  T4DATAMON/CHHOF01P_2  SET  HOFSSYST  =  'TARGET'  WHERE  HOFSSYST  =
'SOURCE'
```

- Run Command:

```
CPYF      FROMFILE(T4DATAMON/CHHOF01P_2)      TOFILE(T4DATAMON/CHHOF01P)
MBROPT(*ADD) CRTFILE(*NO)
```

- Run Command:

```
DLTF T4DATAMON/CHHOF01P_2
```

- Run Command:

```
CPYF      FROMFILE(T4DATAMON/CHDOF01P)      TOFILE(T4DATAMON/CHDOF01P_2)
MBROPT(*ADD) CRTFILE(*YES)
```

- Run SQL statement:

```
UPDATE  T4DATAMON/CHDOF01P_2  SET  DOFSSYST  =  'TARGET'  WHERE  DOFSSYST  =
'SOURCE'
```

- Run Command:

```
CPYF      FROMFILE(T4DATAMON/CHDOF01P_2)      TOFILE(T4DATAMON/CHDOF01P)
MBROPT(*ADD) CRTFILE(*NO)
```

- Run Command:

```
DLTF T4DATAMON/CHDOF01P_2
```

If you were monitoring remote files (with remote journals) on your source system, carry out the following steps. In our examples we assume `SOURCERMT` is the system name for remote files on our source system and `TARGETRMT` is the system name for remote files on our target system (only if `SOURCERMT` and `TARGETRMT` are different):

- Run SQL statement:

```
UPDATE T4DATAMON/CHITP01P SET ITPSSYST = 'TARGETRMT' WHERE ITPSSYST =
‘SOURCERMT’
```

- Run Command:

```
CPYF FROMFILE(T4DATAMON/CHHOF01P) TOFILE(T4DATAMON/CHHOF01P_2)
MBROPT(*ADD) CRTFILE(*YES)
```

- Run SQL statement:

```
UPDATE T4DATAMON/CHHOF01P_2 SET HOFSSYST = 'TARGETRMT' WHERE HOFSSYST =
‘SOURCERMT’
```

- Run Command:

```
CPYF FROMFILE(T4DATAMON/CHHOF01P_2) TOFILE(T4DATAMON/CHHOF01P)
MBROPT(*ADD) CRTFILE(*NO)
```

- Run Command:

```
DLTF T4DATAMON/CHHOF01P_2
```

- Run Command:

```
CPYF FROMFILE(T4DATAMON/CHDOF01P) TOFILE(T4DATAMON/CHDOF01P_2)
MBROPT(*ADD) CRTFILE(*YES)
```

- Run SQL statement:

```
UPDATE T4DATAMON/CHDOF01P_2 SET DOFSSYST = 'TARGETRMT' WHERE DOFSSYST =
‘SOURCERMT’
```

- Run Command:

```
CPYF FROMFILE(T4DATAMON/CHDOF01P_2) TOFILE(T4DATAMON/CHDOF01P)
MBROPT(*ADD) CRTFILE(*NO)
```

- Run Command:

```
DLTF T4DATAMON/CHDOF01P_2
```

- Run SQL Statement:

```
UPDATE T4DATAMON/CHOFA01P SET OFASSYST = 'TARGETRMT' WHERE OFASSYST =
‘SOURCERMT’
```

Tip

When replicating `CHUSR01P` make sure that you check the list of users, to make sure they match the users on the target system.

F.2 Data Areas (*DTAARA)

Replicate the following data area file from the source machine to the target machine. Note that all Data Monitor monitors must be ended in the target machine during replication.

- `CHCONFIG`

F.3 Manual Steps

Finally there are a number of steps that you will have to carry out manually

F.3.1 Authorization list T4DATAMON

Enroll all the users that will be using Data Monitor in the authorization list `T4DATAMON` on the target machine.

F.3.2 Automatic start of Data Monitor monitors

Data Monitor monitors can be configured to start automatically at IPL or when the subsystem `T4DATAMON` starts. As this involves changes in job queues and subsystems descriptions, you must set this option manually.

You can reach the monitor configuration from the Data Monitor main menu (`T4DATAMON/DTM_MENU`):

- For monitoring Files select option **13** start settings
- For monitoring Users select option **16** start settings for monitoring users
- Set the variable Automatic monitor starting to `*YES`.

Appendix G
ALEV Variables and Functions for Expressions

All expressions in Data Monitor are introduced using ALEV scripting language, and must have a valid syntax in order to work correctly. If not, a run-time error will occur, a message will be sent to the product message queue, and the product will assume that no expression has been introduced (as if you had left the field with blanks).

Note
Due to internal limitations, ALEV only supports numeric values up to nine digits. Using numeric values of ten or more digits may result in unpredictable results.

ALEV 1.5 is now supported (replacing ALEV 1.2), improving performance in the execution of ALEV expressions up to 40-50%. ALEV 1.5 is completely compatible with ALEV 1.2, so you do not need to change anything in your existing configuration, but will gain a lot of new functionality. ALEV 1.5 is a scripting language, rather than an expression evaluator (such as ALEV 1.2). It adds standard flow control instructions such as If … Then, For, and While, that give much more power and ease of use for ALEV expressions.

For more details see the *ALEV 1.5 Reference Manual*.

G.1 ALEV Variables

The following table shows the valid ALEV variables and the expressions you can use them in:

Variable	Variable Type	Valid in Store Transaction If of sensitive users, General filter expression of GENJRNDTA (YES/NO)	Valid in Use Format (YES/NO)	Valid in Store Transaction (YES/NO)	Valid in Pre-process (YES/NO)	Available from version
&SourceSystem Name	Char [8]	YES	YES	YES	YES	V5R1
&ReceiverID	Integer	YES	YES	YES	YES	V5R1

Variable	Variable Type	Valid in Store Transaction If of sensitive users, General filter expression of GENJRNDTA (YES/NO)	Valid in Use Format (YES/NO)	Valid in Store Transaction (YES/NO)	Valid in Pre-process (YES/NO)	Available from version
&LocalSystemName	Char [8]	YES	YES	YES	YES	V5R1
&SequenceNumber	Char [20]	YES	YES	YES	YES	V5R1
&KeyDescription	Char [100]	NO	NO	YES	YES	V5R1
&CommitmentControlFlag	Char [1]	YES	YES	YES	YES	V5R1
&CommitCycleID	Zoned (20)	YES	YES	YES	YES	V5R1
&FormatID	Integer	NO	NO	YES	YES	V5R1
&TimeStamp	Char [26]	YES	YES	YES	YES	V5R1
&RecordRRN	Char [20]	YES	YES	YES	YES	V5R1
&JobName	Char [10]	YES	YES	YES	YES	V5R1
&JobUser	Char [10]	YES	YES	YES	YES	V5R1
&JobNumber	Char [6]	YES	YES	YES	YES	V5R1
&ProgramName	Char [10]	YES	YES	YES	YES	V5R1
&ProgramLibrary	Char [10]	YES	YES	YES	YES	V5R2
&FileName	Char [10]	YES	YES	YES	YES	V5R1
&FileLibrary	Char [10]	YES	YES	YES	YES	V5R1
&MemberName	Char [10]	YES	YES	YES	YES	V5R1
&FormatName	Char [10]	NO	NO	YES	YES	V5R1
&RealUser	Char [10]	YES	YES	YES	YES	V5R1
&UserClass	Char [10]	YES	YES	YES	YES	V5R1
&UserGroup	Char [10]	YES	YES	YES	YES	V5R1
&UserAccountingCode	Char [15]	YES	YES	YES	YES	V5R1
&UserCountryIdentifier	Char [2]	YES	YES	YES	YES	V5R1

Variable	Variable Type	Valid in Store Transaction If of sensitive users, General filter expression of GENJRNDTA (YES/NO)	Valid in Use Format (YES/ NO)	Valid in Store Transaction (YES/ NO)	Valid in Pre-process (YES/NO)	Available from version
&IPAddress	Char [46]	YES	YES	YES	YES	V5R2
&PortNumber	Zoned (5)	YES	YES	YES	YES	V5R2
&OperationType	Char [1]	YES	YES	YES	YES	V5R1
&KeyConcatenation	Char [200]	NO	NO	NO	NO	V5R1
&FieldChanged	Char [1]	NO	NO	NO	NO	V5R1
&BeforeBuffer	Char [4096]	NO	YES	YES	YES	V5R1
&AfterBuffer	Char [4096]	NO	YES	YES	YES	V5R1
&BeforeValues	Varchar	NO	YES	YES	YES	V5R1
&AfterValues	Varchar	NO	NO	YES	YES	V5R1
&FieldName	Char [10]	NO	NO	NO	NO	V5R1
&FieldIsKey	Char [1]	NO	NO	NO	NO	V5R1
&OperationValues	Varchar	NO	YES	YES	YES	V5R1
&MonitorSource	Char[1]	YES	YES	YES	YES	V5R1
&Warning	Char[1]	NO	NO	YES	YES	V5R1
&CustomChar1	Char[100]	NO	NO	YES	YES	V5R1
&CustomChar2	Char[100]	NO	NO	YES	YES	V5R1
&CustomChar3	Char[100]	NO	NO	YES	YES	V5R1
&CustomInt1	Integer	NO	NO	YES	YES	V5R1
&CustomInt2	Integer	NO	NO	YES	YES	V5R1
&CustomInt3	Integer	NO	NO	YES	YES	V5R1

G.1.1 Description of the variables:

SourceSystemName: Name of the system where the database operation was carried out.

ReceiverID: Unique numeric identifier assigned by Data Monitor to the receiver where the database operation was stored.

LocalSystemName: Name of the system running Data Monitor and where the database operation was read from journal

SequenceNumber: Sequence number of the database operation stored in journal

KeyDescription: Data Monitor does not fill this variable. You can use it to set a text description for key values of a database operation.

CommitmentControlFlag: Database operation under commitment control. Valid values are:

- Y (Yes)
- N (No)
- R (database operation result of a Rollback).

CommitCycleID: Commit cycle id of a database operation. A value of 0 means that the database operation is not under commitment control.

FormatID: Format ID used to read database operation from journal

TimeStamp: Timestamp of the database operation. Format: YYYY.MM.DD-HH:MM:SS.mmmmmm

RecordRRN: Record relative number of database operation

JobName: Job name used to execute the database operation

JobUser: User name of the job that executed the database operation

JobNumber: Job number used to execute the database operation

ProgramName: Name of the program that executed the database operation.

ProgramLibrary: Program library used to execute the database operation

FileName: File name of the database operation

FileLibrary: File library of the database operation

MemberName: Member name of the database operation.

FormatName: Format name of the database operation

RealUser: Real user who executed the database operation

UserClass: User class of the real user executed the database operation

UserGroup: User group of real user who ran the database operation

UserAccountingCode: Accounting code of the real user who ran the database operation

UserCountryIdentifier: User country identifier of the real user who ran the database operation.

IPAddress: IP address of the remote client that executed the database operation

PortNumber: Local port number of the connection from where the database operation was done

OperationType: Type of database operation. Valid values are:

- A (Add or Insert)
- U (Update)
- D (Delete)
- R (Read)
- C (file Cleared)
- F (file Deleted)
- E (end of journaling for file).

KeyConcatenation: Concatenated key values of database operation

FieldChanged: Field changed in data operation. Valid values are:

- Y (Yes)
- N (No)

BeforeBuffer: Before image buffer of record changed or deleted

AfterBuffer: After image buffer of record changed or added

BeforeValues: Array indexed by field names representing values of fields in record before image.

AfterValues: Array indexed by field names representing values of fields in record after image

FieldName: Field name in database operation

FieldIsKey: Field is a key field for format selected? Valid values are:

- Y (Yes)
- N (No)

OperationValues: equals &AfterValues for Add, Update or Read transactions. For Delete transactions it is the same as the &BeforeValues. You can use this variable instead of &BeforeValues and &AfterValues to simplify Data Monitor for iSeries expressions.

MonitorSource: Type of monitor, which saved the transaction. Valid values are:

- F (Data Monitor for Sensitive Files)
- U (Data Monitor for Sensitive Users)
- O (GENJRNTA, Express)

Warning: Warns about possible errors when saving transaction. Valid values are:

- Blank character (No warning)
- F (Format used could be newer than transaction timestamp)
- E (Decimal data error for some field value)
- O (Format used could be older than transaction timestamp)

CustomChar1: Variable reserved for use in future versions of Data Monitor. The default value is blanks.

CustomChar2: Variable reserved for use in future versions of Data Monitor. The default value is blanks.

CustomChar3: Variable reserved for use in future versions of Data Monitor. The default value is blanks.

CustomInt1: Variable reserved for use in future versions of Data Monitor. The default value is zero.

CustomInt2: Variable reserved for use in future versions of Data Monitor. The default value is zero.

CustomInt3: Variable reserved for use in future versions of Data Monitor. The default value is zero.

G.2 ALEV Functions

The following table shows the valid ALEV functions you can use in any expression. For more information on ALEV see the *ALEV 1.5 Reference manual*.

ALEV Functions
FUNCTION ABS(NUMBER REAL) : REAL
PROCEDURE APPEND(VAR A ARRAY, VALUE VARIANT)
FUNCTION ARCCOS(COSINE REAL) : REAL
FUNCTION ARCSIN(SINE REAL) : REAL
FUNCTION ARCTAN(TANGENT REAL) : REAL
FUNCTION ASSIGNED(VARREF VOID) : BOOLEAN
FUNCTION AVERAGE(V VARIANT, ...) : REAL
FUNCTION BOOLEAN(V VARIANT) : BOOLEAN
FUNCTION COPY(S STRING, INDEX INTEGER, [COUNT INTEGER]) : STRING
FUNCTION COS(ANGLE REAL) : REAL
FUNCTION DATE(D INTEGER, M INTEGER, Y INTEGER): DATE
FUNCTION DATE() : DATE
FUNCTION DATE(V VARIANT) : DATE
FUNCTION DATETIME(V VARIANT) : DATETIME
FUNCTION DAY(D DATETIME) : INTEGER
FUNCTION DAY(D REAL) : REAL
FUNCTION DAY( ) : INTEGER
FUNCTION DAYNAME(D DATETIME) : STRING
FUNCTION DAYOFWEEK(D DATETIME) : INTEGER
FUNCTION DEFINED(VARREF CUSTOM) : BOOLEAN
PROCEDURE DELETE(VAR A ARRAY, KEY_POS VARIANT)
FUNCTION DELETE(S STRING, INDEX INTEGER, [COUNT INTEGER]) : STRING
PROCEDURE EXECMD(CMD_STRING STRING)
FUNCTION EXIST(VARREF CUSTOM) : BOOLEAN
FUNCTION EXP(N REAL) : REAL
FUNCTION FIND(V VARIANT, A ARRAY) : VARIANT
FUNCTION FORMAT(S STRING) : STRING
FUNCTION FRAC(N REAL) : REAL

ALEV Functions
FUNCTION GetCol(LIBRARY STRING, FILE STRING, FIELD STRING, Var SQLCODE INTEGER, [KEYFIELD1 STRING] [KEYVALUE1 Variant], …) : Variant
FUNCTION GetRMTCol(RDBNAME STRING, LIBRARY STRING, FILE STRING, FIELD STRING, Var SQLCODE INTEGER, [KEYFIELD1 STRING] [KEYVALUE1 Variant], …) : Variant
FUNCTION GetRMTRow(RDBNAME STRING, SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)
FUNCTION GetRow(SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)
FUNCTION HEAD(S STRING) : STRING
FUNCTION HOUR(D DATETIME) : INTEGER
FUNCTION HOUR(H REAL) : DATETIME
FUNCTION HOUR( ) : INTEGER
FUNCTION IIF(CONDITION BOOLEAN, IF_TRUE VARIANT, IF_FALSE VARIANT) : VARIANT
FUNCTION INDEXOF(V VARIANT, A ARRAY) : VARIANT
FUNCTION INTEGER(V VARIANT) : INTEGER
FUNCTION ISBLANK(S STRING) : BOOLEAN
FUNCTION ISNULL(V VARIANT) : BOOLEAN
FUNCTION ISNUM(S STRING) : BOOLEAN
FUNCTION ISZERO(N REAL) : BOOLEAN
FUNCTION KEYAT(POS INTEGER, A ARRAY) : VARIANT
FUNCTION KEYOF(V VARIANT, A ARRAY) : VARIANT
FUNCTION LEN(S STRING) : INTEGER
FUNCTION LENGTH(S STRING) : INTEGER
FUNCTION LN(N REAL) : REAL
FUNCTION LOCATE(SUBS STRING, S STRING): INTEGER
FUNCTION LOWER(S STRING) : STRING
FUNCTION LOWERCASE(S STRING) : STRING
FUNCTION LTRIM(S STRING) : STRING
FUNCTION MAX(V VARIANT, ...) : VARIANT
FUNCTION MIN(V VARIANT, ...) : VARIANT
FUNCTION MINUTE(D DATETIME) : INTEGER
FUNCTION MINUTE(M REAL) : REAL
FUNCTION MINUTE( ) : INTEGER
FUNCTION MONTH(D DATETIME) : INTEGER
FUNCTION MONTH(M REAL) : REAL
FUNCTION MONTH( ) : INTEGER
FUNCTION MONTHNAME(D DATETIME) : STRING

ALEV Functions
FUNCTION NOW() : DATETIME
FUNCTION POS(SUBS STRING, S STRING): INTEGER
FUNCTION POSITION(SUBS STRING, S STRING): INTEGER
FUNCTION POW(BASE REAL, EXPONENT REAL) : REAL
FUNCTION RANDOM(...) : VARIANT
FUNCTION REAL(V VARIANT) : REAL
FUNCTION REG(PATTERN STRING) : REGEXP
FUNCTION REGEXP(PATTERN STRING) : REGEXP
FUNCTION REPLACE(S STRING, S1 STRING, S2 STRING) : STRING
FUNCTION REVERSE(S STRING) : STRING
FUNCTION ROUND(N REAL) : INTEGER
FUNCTION RTRIM(S STRING) : STRING
FUNCTION RunRMTSQL(RDBNAME STRING
FUNCTION RunSQL(SQLSTMT STRING, Var SQLCODE INTEGER) : Variant (*)
PROCEDURE SBMEXECMD(CMD_STRING STRING, JOBD_NAME STRING,JOBD_LIB STRING)
FUNCTION SECOND(D DATETIME) : REAL
FUNCTION SECOND(S REAL) : REAL
FUNCTION SECOND( ) : REAL
FUNCTION SELECT(E VARIANT, V0 VARIANT, R0 VARIANT, ...) : VARIANT
FUNCTION SIN(ANGLE REAL) : REAL
FUNCTION SIZEOF(A ARRAY) : INTEGER
FUNCTION SQRT(N REAL) : REAL
FUNCTION STRING(V VARIANT) : STRING
FUNCTION SUBS(S STRING, INDEX INTEGER, [COUNT INTEGER]) : STRING
PROCEDURE SYNEXECMD(CMD_STRING STRING)
FUNCTION TAIL(S STRING) : STRING
FUNCTION TAN(ANGLE REAL) : REAL
FUNCTION TIME(H INTEGER, M INTEGER, [S INTEGER, MS INTEGER]): TIME
FUNCTION TIME() : TIME
FUNCTION TIME(V VARIANT) : TIME
FUNCTION TRIM(S STRING) : STRING
FUNCTION TYPENAME(V VARIANT) : STRING
FUNCTION UPPER(S STRING) : STRING
FUNCTION UPPERCASE(S STRING) : STRING
FUNCTION VARIANCE(V VARIANT, ...) : REAL
FUNCTION WEEKOFMONTH(D DATETIME) : INTEGER

ALEV Functions
`FUNCTION WEEKOFYEAR( D DATETIME ) : INTEGER`
`FUNCTION YEAR( D DATETIME ) : INTEGER`
`FUNCTION YEAR( Y REAL ) : REAL`
`FUNCTION YEAR( ) : INTEGER`

G.3 External Business Functions

In addition to the ALEV variables and functions, Data Monitor for iSeries offers you External Business Functions; functionality that allows you add almost any function or calculation you can imagine to your Data Monitor expressions.

Previously, you could only use the ALEV variables and functions defined earlier in this appendix to create expressions for Data Monitor to decide when and how to

- use formats
- store transactions
- pre-process data
- audit transactions
- save fields

However, the data you could configure in these expressions was limited to information retrieved previously by Data Monitor or data stored in the monitored files. To be more specific Data Monitor can retrieve two different sets of variables:

- Information or data provided by Data Monitor, such as
  - `&SourceSystemName`
  - `&IPAddress`
  - `&RealUser`
  - `&KeyConcatenation`
  - etc.
- Data stored in fields of the monitored databases, such as
  - `&BeforeValues`
  - `&AfterValues`
  - `&OperationValues`
  - etc.

In addition to the variables mentioned here, Data Monitor for iSeries includes the following ALEV functions:

- `EXECMD`,
- `SBMEXECMD` and
- `SYNEXECMD`

External Business Conditions allow you to work with values returned from any existing functions or procedures on your system, in addition to the variables and functions mentioned above. But what is an external business function? An external business function is a procedure or function that the user can define, implement and integrate in Data Monitor for iSeries. This procedure or function is part of a service program which you can include in Data Monitor for use in Data Monitor expressions. External business functions are very powerful as you can include any function or procedure in the service program for use in Data Monitor. With external business functions you can retrieve any value on your system or even on other systems and use it in any of your Data Monitor expressions. You are only limited by the scope of your programs. This is why external business function is such a powerful feature. For example, you can retrieve a value derived from complex business calculations (such as interest rates, mortgage payments, sales reps commissions, and more) that you have in an RPG program.

Note

If the values you want to retrieve are simply the values of any field in any record of any file, you should better use ALEV functions for 360º enrichment as for example `GetRow`.

G.3.1 Alternative way to run external programs and retrieve results

You can use ALEV functions for 360º enrichment in the following way:

- First use ALEV function `RunSQL` to run a `CALL SQL` Statement, which calls your program with the business logic you want to reuse. This program must leave the information you want to retrieve in any database table in your system.
- Then, you just have to use ALEV functions `GetRow` or `GetCol` to access the information in the previous database tables.

Note

This way is easier to implement than external business functions but if you have performance concerns, especially if your business logic has to be called lots of times, using external business functions can be much faster.

Depending on the efficiency of the programs you include, or the number of calls made to a program, you may experience reduced performance in Data Monitor. This potential reduction in performance is entirely due to the performance of the external program and/or the way it is implemented. We recommend you only use external business functions when they are absolutely necessary and that you carefully consider these issues when creating them.

Imagine that the external business function you use in an expression requires a lot of time to process a result. Data Monitor will have to wait for this result before it can decide, for example, whether to use a particular format, or audit a particular transaction. It would then appear that there is a reduction in the performance of Data Monitor, whereas it is really the external program that is at fault. We hope this example illustrates that it is important to select your scenarios with care, carefully consider when to use an external business function and what the effects will be on Data Monitor.

When working with external business functions we recommend you execute the four required steps in the following order:

Step 1. If you do not yet have a Service Program with the functions you want to use in Data Monitor, create one first.

Step 2. Include the functions in the Data Monitor environment

Step 3. Create the desired expressions in Data Monitor

Step 4. Start or restart the monitors.

Important

Currently you can define up to 12 external functions. Data Monitor provides another three conditions by default: `EXECMD`, `SBMEXECMD` and `SYNEXECMD`. These conditions are part of the product and should not be changed.

G.3.2 Example of an External business function

Retail banks check a customer’s credit balance to decide whether or not to allow a credit card transaction. Likewise, they check the client’s bank balance before issuing cash from an ATM. These types of operational checks are completed by the retail banking application and form part of the bank’s daily operations.

However, although authorization for an individual transaction may be legitimate, a series of suspicious transactions might point to potential problems with a debit or credit card. For example, large amounts of money drawn from different ATMs in a short space of time, or an unusual series of purchases executed abroad, might indicate that customer’s card has been stolen or is being used for money laundering purposes. Of course, the client may simply have gone on a holiday spending spree, or need cash to pay for their lodgings. Either way, the bank’s Risk Management department will need to be alerted to such series of events and follow up on the transactions with their clients to limit risk in the event of illegitimate use of the cards. Frequently, these events come to light when several subsequent transactions are rejected.

Therefore the bank in our example will implement business rules to start auditing client accounts when the amount of the requested transaction exceeds the customer’s available credit or balance. Data Monitor can retrieve a wide range of data from various sources, including the retail banking application, or the audited transactions, such as the amount and location of the transaction, the customers contact details, and more. This data can later be used in reports to help Risk Management determine potential problems and limit damages.

In our example the bank uses the retail banking application to retrieve the customers account balance. The entry point of this application is a program called `T4RUA01C`. This program retrieves the account balance from the database `USRBLC01P`. The DDS file for the database is as follows:

```
    Customer account balance file USRBLC01P. This file contains
    the customers’ account balance in Euros.


  A    UNIQUE

  A    CCSID(65535)

  A    R USRBLC01F                                    *

  A    USRNAME       10     COLHDG('User Name')

  A    USRBLCN       09B00  COLHDG('Balance')        *

  A    K USRNAME
```

In order to use the program `T4RUA01C` in Data Monitor expressions:

Step 1. Compile the source code `T4EXTCNDI` and create the corresponding service program as described in *section G.3.5 - Creating the Service Program on page 127*.

Step 2. Insert the function GetUserBalance in the list of external business functions in Data Monitor. For details *see section G.3.6 - Defining functions or procedures in Data Monitor for iSeries on page 129.*

G.3.3 How can you adapt your programs to Data Monitor for iSeries environment?

Data Monitor for iSeries provides you with an example of code for an interface program to show you how to build or integrate your external business functions. You can use this program code to create your function either by directly entering the required code in the body or by calling an existing external program or service program on your system.

This code is written in Standard C language. All the necessary sample source code is included in the installation and you can find it in the files H and QCLESRC in the T4NICELINK product library. The source T4EXTCNDI contains the code for integration of your functions in ALEV and the necessary include files for compiling and creating your service programs. In addition you will find a file called QSRVSRC in the T4NICELINK library, which you can use to export a specific set of functions from your service program.

Note that this example represents only one of the many ways you can integrate your external business functions. The example presented here is intended only to help you understand how to integrate your functions in Data Monitor. You are not obliged to use this code to integrate your business functions.

To keep the example simple and easy to follow, we have used only the most basic elements required for integrating a function. Of course ALEV is much more powerful and offers a wide range of variables and data structures. To get a better idea of the possibilities when working with ALEV, have a look at the <alev.h> include file.

We will continue to discuss the content of the example's source code available in T4EXTCNDI. You will find a number of highlighted sections which will be discussed in more detail below.

```
/* Constants */


#define  NUM_REQ_PARMS_USER 1


/* System Includes */


#include <stdlib.h>

#include<stdio.h>

#include<recio.h>

#include <qusec.h>          /* Error code Structures        */

#include <except.h>         /* Exception handler structures */

#include<string.h>

#include <decimal.h>

#include<qcmdexc.h>
```

Section A

```
/* User Includes */
#include <alev.h>            /* ALEV functions               */
#include <err_msg.h>         /* ALEV error messages          */
#include "t4extcndi.h"       /* Prototype definitions        */
#include "t4rua01c.h"        /* Retrieve User Balance        */
```

```
/*----------------------------------------------------------*/
/* DefineFunction("FUNCTION GetUserBalance(USERNAM
STRING):INTEGER",NULL);                                     */
/* Return valid values:                                     */
/*  0:  No error                                            */
/* -1:  Incorrect number of parameters                      */
/* -2:  Incorrect parameter list                            */
/*----------------------------------------------------------*/
```

Section B

```
long int GetUserBalance(TStdArgumentList * ArgList, void ** Result){
```

```
   /* Local variables */
```

Section C

```
   /* Pointers to ALEV argument items */
   TStdValue               *ArgValue;
   TStdArgument            *ArgItem;
```

```
   /* Specific functions parameter values  */
   char                    UserNam[10];
   /* Variables to work with output value in external bound program */
   int                     AuthNbr;
   /* Error variables to work with error code in external bound
   program */
   char                    error_code[2];
```

```
/* BODY */

```

Section D

```
/* Initialize result type and value */
PubResult.Kind      = typeInteger;
PubResult.AsInteger = -1;

/* Initialize local variable value */
memset(UserNam, ' ', sizeof(UserNam));

```

Section E

```
/* Evaluate input parameters */
if (ArgList == NULL){
   *Result = &PubResult;
   return ERR_INPUT_PARAMETERS;
}/*if*/

/* Validate the expected number of parameters for this function */
if (ArgList->fArgCount != NUM_REQ_PARMS_USER){
   *Result = &PubResult;
   return ERR_NUMBER_OF_PARAMETERS;
}/*if*/

/* Analyze parameter items */

```

Section F

```
/* Pointing at the first parameter in the input parameter list */
ArgItem = ArgList->fFirst;
if (ArgItem == NULL) return ERR_INPUT_PARAMETERS;
ArgValue = ArgItem->Value;
if (ArgValue == NULL) return ERR_INPUT_PARAMETERS;

/* Get the value. It depends on the parameter type. */
if ((ArgValue->Kind) != typeString) return ERR_INPUT_PARAMETERS;
memcpy(UserNam, ArgValue->AsString, strlen(ArgValue->AsString));
```

Section G

```
   /*In our case, an external RPG program to retrieve from the
   database */

   Rtv_UserBalance(UserNam,&AuthNbr,error_code);

   if (strncmp(error_code, "  ",2) == 0) PubResult.AsInteger =
   AuthNbr;
```

Section H

```
   /* Return result */

   *Result = &PubResult;

   return 0;


}/* GetUserBalance */
```

Tip

To simplify this example we've used external functions to retrieve the data we need. The external functions may be programmed in RPG, COBOL, C, C++, CL, Java, or any other programming language of your choice.

G.3.4 Description of Sections

Section A

The first include file is necessary to declare the functions in this sample source code.

```
/* User Includes */

#include <t4extcndi.h>               /* Function prototype  */
```

The t4extcndi.h include file defines the variable PubResult as the output parameter which will return the value of the function.

```
#ifndef _t4extcndi_h

#define _t4extcndi_h

#define ERR_NUMBER_OF_PARAMETERS    -1

#define ERR_INPUT_PARAMETERS        -2

long int  GetUserBalance(TStdArgumentList * ArgList, void ** Result);

/* Global variables */

TStdValue                           PubResult;

/* Result value */

Char                                FuncErrorMsg[1024];

#endif
```

In this example the output will be an integer value representing the customer's account balance or credit balance. Note that PubResult is declared as TstdValue. For more details regarding this type of variable see the data structure in the include file alev.h.

```
/* Value Data Structure------------------------------------*/
typedef struct _StdValue { TType          Kind;     /* Value Type */
unsigned char  AsBool;  TStdReal       AsReal;
long int       AsInteger;  TStdDateTime   AsDateTime;
char          *AsString;  TStdSetItem  *AsSet;  TStdArray   *AsArray;
void           *AsVoid;};
```

The next two include files are necessary to work with ALEV. The first file <alev.h> provides us with the data structures, variables, functions, etc. that we need when integrating the external programs into ALEV. The second file <err_msg.h> contains potential error messages.

```
#include <alev.h>                  /* ALEV functions      */
#include <err_msg.h>               /* ALEV error messages */
```

The include file t4rua01c.h is required if you want to work with other external programs, modules, etc. In our example, we use one external module that contains one ILE CL module and another ILE RPG module to retrieve the data from the databases.

```
#include "t4rua01c.h"             /* Retrieve User Balance */
```

Depending on whether you want to call external programs from the T4EXTCNDI module you will need different include files. In this particular case we want to call the external program T4RUA01C in library BALANCE. However, we want to map it using the name using Rtv_UserBalance. To achieve this you need the following include file:

```
#ifndef _t4rua01c_h
#define _t4rua01c_h
void Rtv_UserBalance(char UserName[10], int *UserNbr, char
error_code[2]);
#pragma map(Rtv_UserBalance,"BALANCE/T4RUA01C")
#pragma linkage(Rtv_UserBalance, OS, nowiden)
#endif
```

If you are familiar with C you can skip to section B.

You can use #pragma argument to identify a function as an external program under OS/400. In our example we use use #pragma map to map the qualified program name T4RUA01C in library BALANCE as Rtv_UserBalance. The module T4EXTCNDI will then call the program with this name.

If you need to call a bound procedure, for example a CL ILE module called also T4RUA01C, the include file will be as follows:

```
…
#pragma map(Rtv_UserBalance,"T4RUA01C")
#pragma argument(Rtv_UserBalance, OS, nowiden)
…
```

Section B

Function `GetUserBalance`. The first parameter `ArgList` is the list of arguments for the function or procedure. The second parameter Result is the return value for this function. In our case we expect one string parameter representing the real user. And the return value will be an integer representing the balance for that user.

Section C

These are the variables we use to manage the list of input variables.

`TStdArgument` is the pointer to the argument and `StdValue` is the value for the corresponding argument. The value type was mentioned in Section A. Both types are explicitly defined in `alev.h`. The following code represents the argument items in the list:

```
/* Function Arguments-----------------------------------*/
typedef struct _StdArgument {TStdValue   *Value;  /* Argument Value */
               TStdArgument   *Next;    /* Next Argument in the list */
                };
```

Note
Data Monitor does not distinguish between service programs and library names. Therefore, you can not configure two functions with the same prototype even if they belong to different service programs.

Section D

In our example the result is an integer value according to the internal authority number for the users in the database. We initialize this variable with the value -1, as we do not consider this a valid value for this parameter.

As you can see we have set the value using the `AsInteger` data field in the `TstdValue` data structure mentioned in Section A and set the type of variable to integer using the Kind data field and `typeInteger` as one of the valid values for ALEV.

Here follow the valid values. You can also find this information in the file `alev.h`.

```
/* Types of Data---------------------------------------*/
typedef enum    { typeBoolean = 0,      /* Boolean Type */
                  typeInteger,          /* Integer Types */
                  typeReal,             /* Real Types */
                  typeTime,             /* Time Type */
                  typeDateTime,         /* DateTime Type */
```

```
                    typeString,          /* String Type */

                    typeSet,             /* Set Type */

                    typeNULL,            /* Variant Type */

                    typeArray,           /* Array Type */

                    typeVoid,            /* Void Type */

                    typeRegExp,          /* Regular Expression Type */

                    typeChar,            /* Character */

                    typeItemRange        /* Item's range */

                                  } TType;
```

Section E

This part of the code evaluates the list of parameters. If they are not correct or not expected, it will return an error. In this section we evaluate the input parameter values. If there is any problem with the input argument list a value different to 0 is returned indicating that the function couldn't finish correctly.

For example, first we evaluate whether the number of input parameters in the argument list is as expected. We use the standard argument data structure defined in alev.h, as follows:

```
typedef struct  _StdArgumentList{

                TStdArgument *fFirst;    /* First Argument, if any */

Int             fArgCount;               /* Number of Arguments */

                };
```

The field fArgCount is used to find the number of parameters in the list of arguments. The field fFirst is a pointer to the first argument item in the list.

Section F

Working with the list of parameters we get the value for the Real User Name. With this value we call the program that will retrieve available credit or balance for this customer. To retrieve the value we need to use the fFirst field for argument list data structure mentioned in the previous section and the Value field from the StdValue data structure mentioned in Section D. Furthermore, we need the Kind field from the StdValue data structure to check the argument type we're currently using.

Section G

This section describes how to retrieve the value from an external module.

The function Rtv_UserBalance can be implemented as an external program or its code can be directly included in the same service program. This function can be implemented in your preferred programming language. Here we have added it to the service program as an external program. You will find the include file definition of t4rua01c.h in Section A.

For more information regarding defining external programs or bound procedures see the discussion of section A.

In our example the program `Rtv_UserBalance` is actually the program `T4RUA01C` in library `BALANCE`. This program retrieves the balance for a specified user. The program uses an ILE CL module as a PEP (Program Entry Procedure) and an ILE RPG module to retrieve the data from the database.

Section H

Return the value in the variable Result. If this function returns a value different to 0 then you will find an ALEV error message in the product message queue. To open the product message queue in Data Monitor press F9 from the Working with Monitored Files menu.

G.3.5 Creating the Service Program

Once you have adapted your programs, the next step is to create the service program. Later, you will use the service program to define your functions or procedures in the list of external business function s in Data Monitor for iSeries.

`T4EXTCNDI` implements a function called `GetUserBalance`. This is the function we want to export in the service program. For purposes of our example we will create a service program named `GETBALSRV` in library `BALANCE`.

There are many ways of creating a service program. Although we assume that the person using external business functions already has the knowledge and necessary tools to create a service program, we will briefly mentions the required steps:

Step 1. Compile all the elements.

In our example you only need to compile the module `T4EXTCNDI`. This module is located in `T4DATAMON/QCLESRC`. Data Monitor for iSeries provides you with the required include files for compiling module `T4EXTCNDI`.

Here we assume that you want to create your service program in the library called `BALANCE`. We further assume that the program `T4RUA01C` is also located in the library `BALANCE`.

To compile the module run the following commands:

```
ADDLIBLE LIB(T4NICELINK) POSITION(*FIRST)
```

```
CRTCMOD MODULE(BALANCE/T4EXTCNDI) SRCFILE(T4NICELINK/QCLESRC)
SRCMBR(*MODULE) OUTPUT(*PRINT) DBGVIEW(*ALL) REPLACE(*YES)
TGTRLS(*CURRENT)
```

Step 2. To create and grant authorities for the service program `GETBALSRV`, execute the following commands:

```
CRTSRVPGM SRVPGM(BALANCE/GETBALSRV) MODULE(BALANCE/T4EXTCNDI)
EXPORT(*SRCFILE) SRCFILE(T4NICELINK/QSRVSRC) SRCMBR(*SRVPGM)
ACTGRP(*CALLER)
```

```
CHGOBJOWN OBJ(BALANCE/GETBALSRV) OBJTYPE(*SRVPGM) NEWOWN(QPGMR)
```

```
RVKOBJAUT OBJ(BALANCE/GETBALSRV) OBJTYPE(*SRVPGM) USER(*PUBLIC)
AUT(*ALL)
```

```
GRTOBJAUT OBJ(BALANCE/GETBALSRV) OBJTYPE(*SRVPGM) USER(*PUBLIC)
AUT(*USE)
```

Note
When creating the service programs and the corresponding libraries please take into account that Data Monitor for iSeries needs `*USE` authority over these objects.

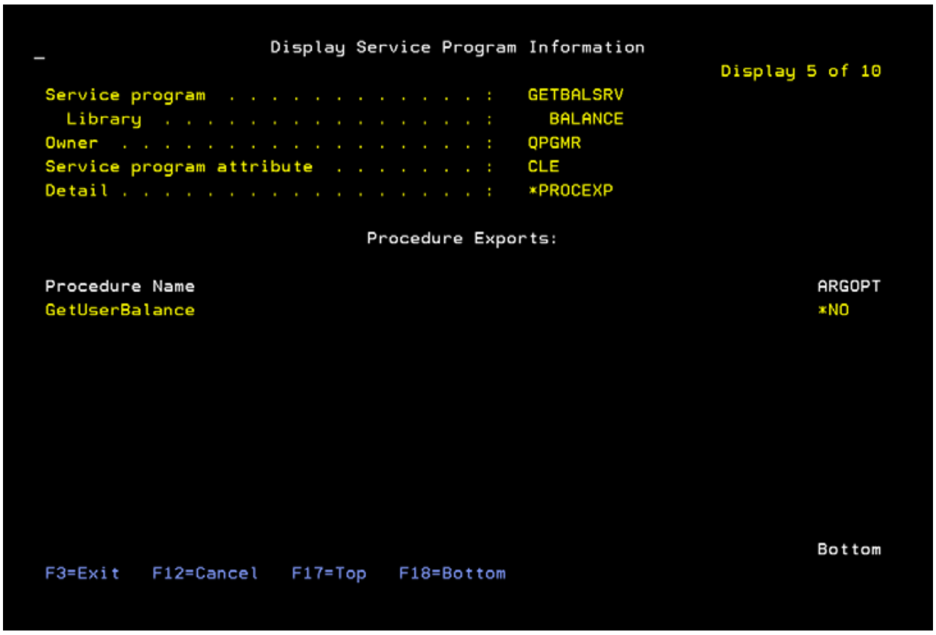
Once you've executed these commands you will see the service program `GETBALSRV` and a new module `T4EXTCNDI` in your library `BALANCE`.

If you want you can now delete the module `T4EXTCNDI`.

Step 3. We recommend you check that the function `GetUserBalance` has been correctly exported by running the command

```
DSPSRVPGM SRVPGM(BALANCE/GETBALSRV) DETAIL(*PROCEXP)
```

The following screen should appear:



```
                        Display Service Program Information
_                                                            Display 5 of 10
 Service program  . . . . . . . . . . . . :   GETBALSRV
   Library  . . . . . . . . . . . . . . . :     BALANCE
 Owner  . . . . . . . . . . . . . . . . . :   QPGMR
 Service program attribute  . . . . . . . :   CLE
 Detail . . . . . . . . . . . . . . . . . :   *PROCEXP

                               Procedure Exports:

 Procedure Name                                                       ARGOPT
 GetUserBalance                                                       *NO









                                                                      Bottom
 F3=Exit    F12=Cancel    F17=Top    F18=Bottom
```

Figure 85 – Display service program information

In our example we described a service program containing one function that calls an external program. However, there possibilities are endless. You could for example create a service program with more functions, more modules, calling procedures in the same service program or in other service programs, and more.

Depending on the how you create your service program, the include files for the functions must be changed to correctly compile the module `T4EXTCNDI` and create the service program as required. For example, if you want to use the sample code provided here to call existing service programs on your system, you could first create a wrapper program to call these existing service programs. Alternatively, you could change the include files in our example and create a service program that calls other service programs.

The next step is to include `GetUserBalance` in the external business functions list in Data Monitor for iSeries, as described in the following section.

G.3.6 Defining functions or procedures in Data Monitor for iSeries

You can configure functions and procedures correctly exported in your service programs using the configuration file T4EFD01P in library T4DATAMON. The DDS (Data Description Specifications) file for T4EFD01P is:

```
A         UNIQUE
A         CCSID(65535)
A         R T4EFD01F                                            *
A         SRVPGMN         10      COLHDG('Srv. Pgm. Name')
A         SRVPGML         10      COLHDG('Srv. Pgm. Library')
A         FNCTNAM         50      COLHDG('Function Name')
A         FNCPROT         256     COLHDG('Function Prototype')
A         FNCDESC         50      COLHDG('Function Description')
A         ENABDIS         1       COLHDG('Enabled/Disabled')
A         HIDEUNH         1       COLHDG('Hide/Unhide')          *
A         K SRVPGMN
A         K SRVPGML
A         K FNCTNAM
A         K FNCPROT
```

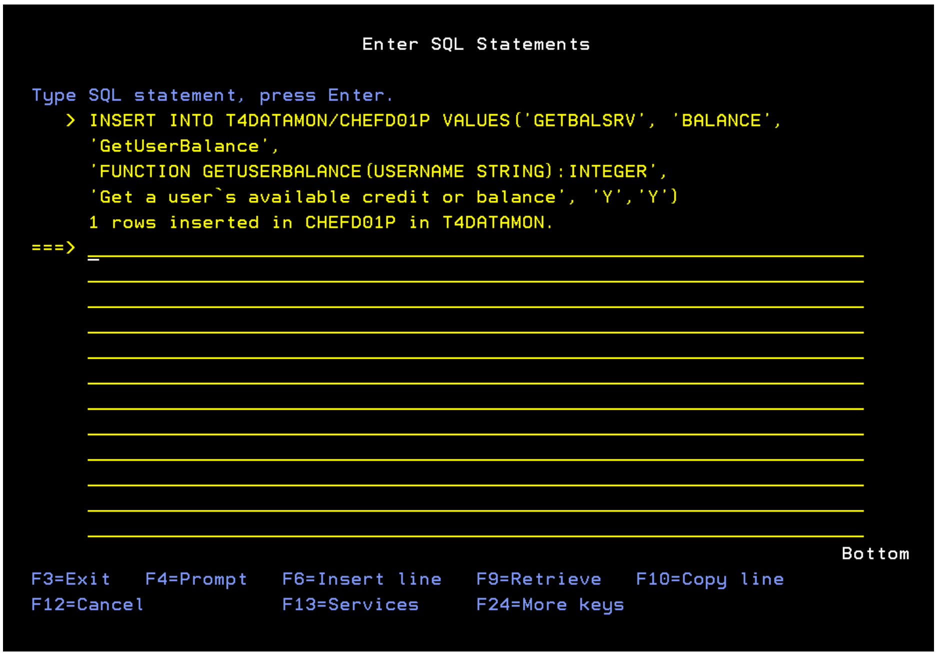
Description fields

- SRVPGMN: The service program name where the functions you want to define are correctly exported.
- SRVPGML: Service program library.
- FNCTNAM: Contains the name of the function you want to work with in Data Monitor expressions. In our example we will use GetUserBalance.
- FNCPROT: This is the prototype for the function or procedure with which you want to work. For example, for function GetUserBalance the prototype would be FUNCTION GETUSERBALANCE(USERNAM STRING):INTEGER. For additional information regarding the format of prototypes *see section G.2 - ALEV Functions on page 114*. Although it is not mandatory to write the format in upper case, we recommend you use capitals for coherence with the other implemented functions.
- FNCDESC: This is the description of the procedure or function you are defining. It helps the user to identify easily the external business functions defined in Data Monitor for iSeries.
- ENABDIS: This flag is used to enable or disable the entry in Data Monitor. If this flag is disabled Data Monitor will ignore this external business function and it will not be integrated in Data Monitor.
- HIDEUNH: With this internal flag you determine whether to show the user the configuration entry. In this version it must always be enabled; Always enter “Y” for this value.

In our example, we consider that the function `GetUserBalance` is exported in the service program `GETBALSRV`, which is located in Library `BALANCE`.

To complete the configuration you must enter this data in the external business function configuration file in Data Monitor. In our example you must insert a record in the `T4NICELINK/T4EFD01P` configuration file. You can achieve this in a 5250 session using the `STRSQL` command. Other alternatives are to use the command `UPDDTA`, ODBC or any other method for inserting a record correctly in the configuration file.

In our example, we use the command `STRSQL` from a 5250 session on the iSeries. Then we use the SQL command `INSERT` as is showed in the next figure:



```
                          Enter SQL Statements

Type SQL statement, press Enter.
   > INSERT INTO T4DATAMON/CHEFD01P VALUES('GETBALSRV', 'BALANCE',
     'GetUserBalance',
     'FUNCTION GETUSERBALANCE(USERNAME STRING):INTEGER',
     'Get a user`s available credit or balance', 'Y','Y')
     1 rows inserted in CHEFD01P in T4DATAMON.
===> _

                                                                   Bottom
F3=Exit   F4=Prompt   F6=Insert line   F9=Retrieve   F10=Copy line
F12=Cancel            F13=Services     F24=More keys
```

Figure 86 – Insert the GetUserBalance function in Data Monitor's external business function configuration.

Here enter the command:

```
INSERT INTO T4DATAMON/CHEFD01P VALUES(

‘GETBALSRV’,

‘BALANCE’,

‘GetUserBalance’,

‘FUNCTION GETUSERBALANCE(USERNAM STRING):INTEGER’,

‘Get the internal authority number of a user’,

‘Y’,

‘Y’)
```

This completes the configuration of the external business functions in Data Monitor for iSeries.

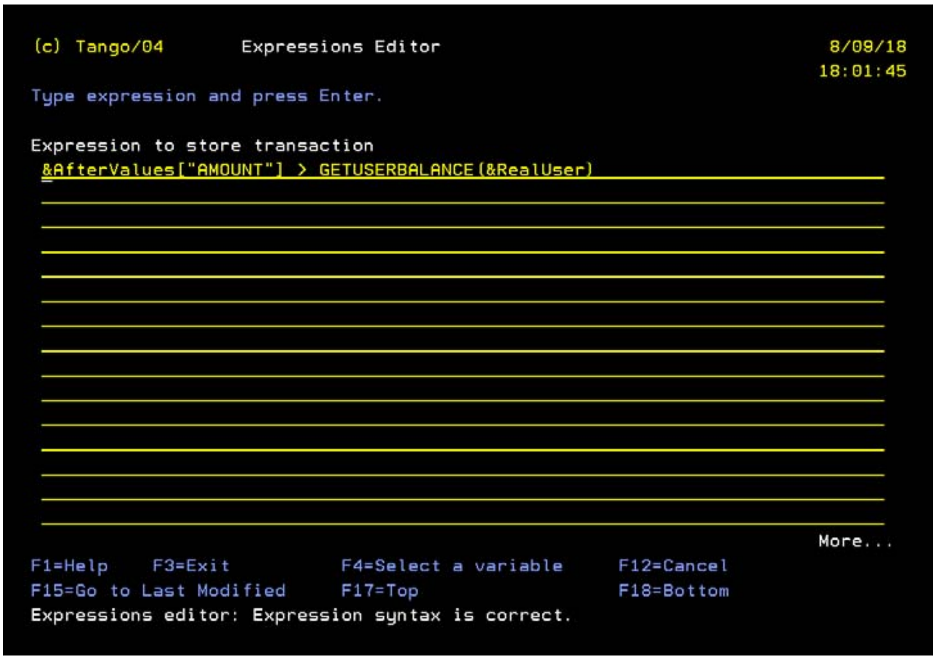
However there is one last step. For the new configuration of external business functions to take effect you must restart the monitor. Please end the monitor and then start the monitor to use the new configuration.

Note that it is not necessary to restart the monitor if you are making changes to an existing external business function, only if you are adding a new function.

G.3.7 Using external business functions

Now you are ready to include the external business functions. Once you have started the monitors again you can work with these new functions in your expressions.

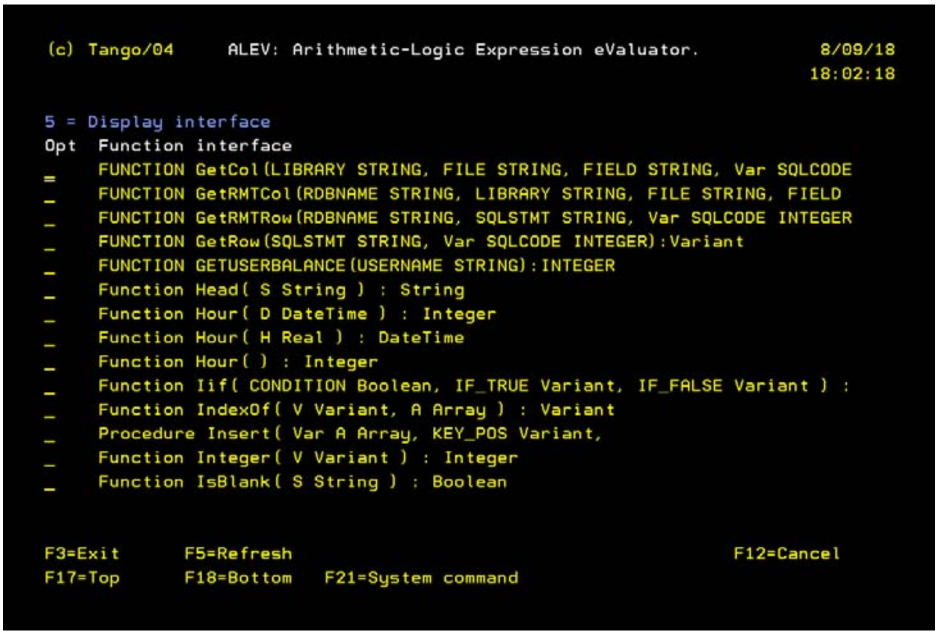
Imagine you want to audit a customer transaction only if the amount of the requested transaction exceeds the customer’s available credit or account balance, which is retrieved from a record in another database. You can add an external business function to your *store transaction if* expression. The required expression is shown in the following image:



```
(c) Tango/04       Expressions Editor                                  8/09/18
                                                                      18:01:45
Type expression and press Enter.

Expression to store transaction
 &AfterValues["AMOUNT"] > GETUSERBALANCE(&RealUser)

                                                                     More...
F1=Help    F3=Exit          F4=Select a variable      F12=Cancel
F15=Go to Last Modified     F17=Top                   F18=Bottom
Expressions editor: Expression syntax is correct.
```

Figure 87 – External business functions in a “store transaction if” expression

You can check the prototype of the new functions or procedures using the help (F1 function). The following image shows you how the functions are defined.



```
 (c) Tango/04     ALEV: Arithmetic-Logic Expression eValuator.          8/09/18
                                                                       18:02:18

 5 = Display interface
 Opt  Function interface
 _    FUNCTION GetCol(LIBRARY STRING, FILE STRING, FIELD STRING, Var SQLCODE
 _    FUNCTION GetRMTCol(RDBNAME STRING, LIBRARY STRING, FILE STRING, FIELD
 _    FUNCTION GetRMTRow(RDBNAME STRING, SQLSTMT STRING, Var SQLCODE INTEGER
 _    FUNCTION GetRow(SQLSTMT STRING, Var SQLCODE INTEGER):Variant
 _    FUNCTION GETUSERBALANCE(USERNAME STRING):INTEGER
 _    Function Head( S String ) : String
 _    Function Hour( D DateTime ) : Integer
 _    Function Hour( H Real ) : DateTime
 _    Function Hour( ) : Integer
 _    Function Iif( CONDITION Boolean, IF_TRUE Variant, IF_FALSE Variant ) :
 _    Function IndexOf( V Variant, A Array ) : Variant
 _    Procedure Insert( Var A Array, KEY_POS Variant,
 _    Function Integer( V Variant ) : Integer
 _    Function IsBlank( S String ) : Boolean


 F3=Exit      F5=Refresh                                        F12=Cancel
 F17=Top      F18=Bottom    F21=System command
```

Figure 88 – External business functions prototypes

Important

When adding functions from the interface to an existing monitor, it is important to follow the steps detailed in the introduction of *section G.3 - External Business Functions on page 117*. Otherwise, you may encounter unexpected results in the evaluation of your expressions or find errors in the product message queue.

The same occurs when working with the Data Monitor expressions editor in interactive sessions. That is, if you are working with a specific set of external business functions and you add one or more external business functions you need to restart the monitors and sign on to a new 5250 session in order for the new functions to work.

From now on the active monitors will use the external business functions to make decisions. You can use external business functions in all Data Monitor expressions, including what format to use, what transactions to store and what fields to save, or to set values using preprocess.

G.3.8 Final considerations

External business functions open a new perspective regarding the data you can handle in your expressions. Before you could only manage the data coming from Data Monitor or the monitored files. Now, you can work with any data you need. It depends on the goals of your programs. For example, you could need to retrieve data from other systems or a system value, and more.

It is important that you thoroughly analyze when to use External business functions to achieve your goals and what is the most efficient way of applying them in Data Monitor. Although external business functions are extremely powerful, they may also result in decreased performance. To be specific, if your monitored file contains a lot of transactions and your program needs a lot of time to return a result, Data Monitor will not be able to evaluate the expression in a short time. As a result you may notice a decrease in performance due to the time your programs need.

The real power of this new feature lies in the wide range of possibilities that you can program. Enrich data with info from other system objects or unmonitored databases.

Imagine that Data Monitor monitors a file containing your customer's bank account number(s) and that the bank account number includes a branch code. You can enrich this data for reporting purposes with additional information stored on your system but outside the scope of Data Monitor.

Suppose we want to use the bank account number to find the following additional information

- Name(s) of the account holder(s)
- Bank name
- Branch address

The bank name and address are stored in a table on your system, but outside the scope of Data Monitor. The names of the account holders can be retrieved from a different database provided by the bank and stored on your system.

This information can be retrieved using a preprocess expression. For example we can use the LOCKEYD (List of Changes Key Description) field, which is a variable length field, to store the additional data. Later you can use the information in reports.

This is only an example of information you could retrieve. The possibilities are endless.

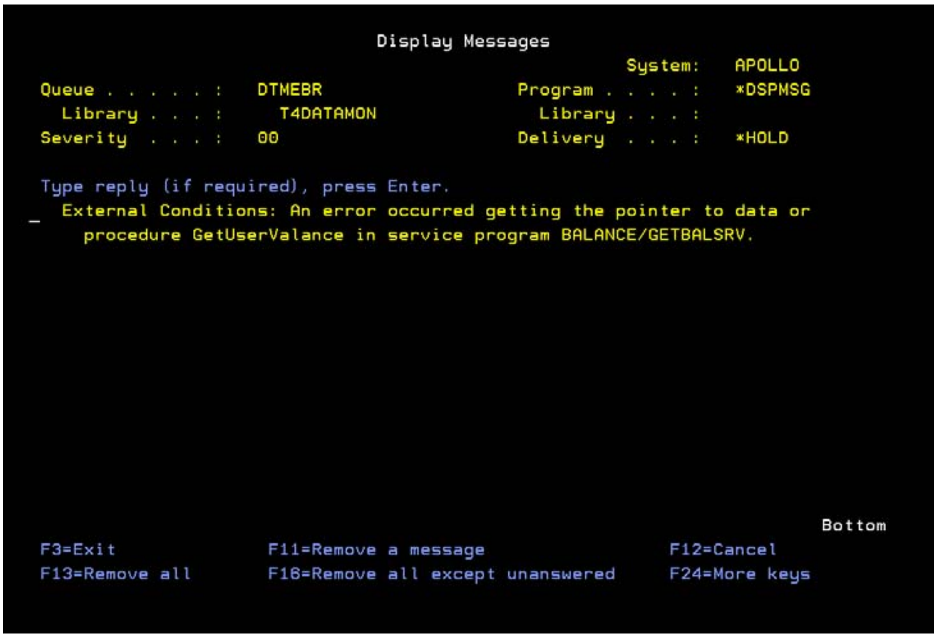
G.4 Troubleshooting

When you encounter a problem we recommend you check the error message queues and product message queues to make sure your external business functions have been included correctly in Data Monitor.

What happens if you don't enter the real name of an exported function in a service program?

When you start the monitors they try to load all the exported functions you defined in the configuration file T4DATAMON/CHEFD01P. However, if an error was made when introducing the real name of an exported function, Data Monitor will not be able to load it. You will find an error message in the message queue T4DATAMON/DTMEBR and in the operator message queue QSYSOPR.

The following image shows the error message:



```
                              Display Messages
                                                    System:   APOLLO
 Queue . . . . . :   DTMEBR                 Program . . . . :   *DSPMSG
   Library . . . :     T4DATAMON              Library . . . :
 Severity  . . . :   00                     Delivery  . . . :   *HOLD

 Type reply (if required), press Enter.
   External Conditions: An error occurred getting the pointer to data or
_    procedure GetUserValance in service program BALANCE/GETBALSRV.










                                                                     Bottom
 F3=Exit          F11=Remove a message               F12=Cancel
 F13=Remove all   F16=Remove all except unanswered   F24=More keys
```

Figure 89 – Error message

In this figure the error appears because the `GetBalance` function you configured does not exist in the service program `GETBALSRV` in library `BALANCE`. As explained earlier, the correct name for the function is `GetUserBalance`.

Although the incorrectly defined functions are not loaded, Data Monitor will continue to load any other correctly defined functions. The monitors will work correctly with the current and valid functions Data Monitor was able to load.

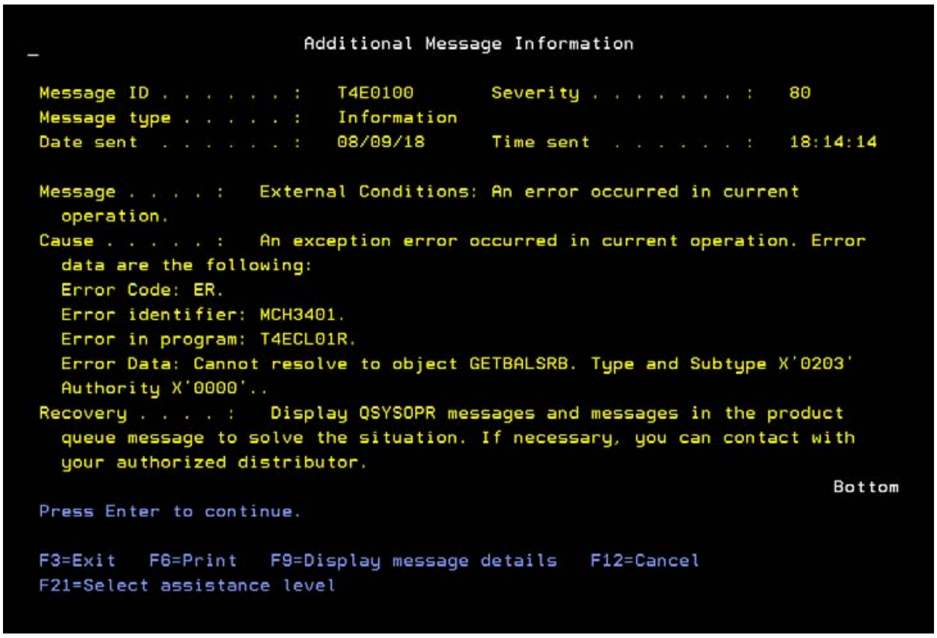
Check the error messages to correct the functions that were incorrectly defined in your external business functions configuration.

Data Monitor shows the same behavior when you use the interface to create an expression with an External Business Function.

What happens if you enter an invalid or non-existent service program name or library?

An exception error message will be sent to `T4NICELINK/T4ECLLOG` message queue and `QSYSOPR` indicating the invalid name for the service program name or library you've introduced. You need to correct doing an update of the record you introduced previously.

This figure shows the error as it appears in Data Monitor:



```
                         Additional Message Information
_
 Message ID . . . . . . :   T4E0100        Severity . . . . . . . :    80
 Message type . . . . . :   Information
 Date sent  . . . . . . :   08/09/18       Time sent  . . . . . . :    18:14:14

 Message . . . . :   External Conditions: An error occurred in current
   operation.
 Cause . . . . . :   An exception error occurred in current operation. Error
   data are the following:
   Error Code: ER.
   Error identifier: MCH3401.
   Error in program: T4ECL01R.
   Error Data: Cannot resolve to object GETBALSRB. Type and Subtype X'0203'
   Authority X'0000'..
 Recovery . . . . :   Display QSYSOPR messages and messages in the product
   queue message to solve the situation. If necessary, you can contact with
   your authorized distributor.
                                                                      Bottom
 Press Enter to continue.

 F3=Exit   F6=Print   F9=Display message details   F12=Cancel
 F21=Select assistance level
```

Figure 90 – Exception error message

Whether Data Monitor is able to audit the expression depends on several factors. For instance it depends on whether the expression is a preprocess expressions or if it used to decide whether to audit transactions or save fields. In all cases you may find unexpected results.

What happens if you enter a prototype with a different function name to that in your service program?

When entering functions in Data Monitor you can assign a different function name from that of the function in the Service Program. Once you have defined a different name in Data Monitor it is important that you use this name in your expressions, not the function name in your Service Program. To find the prototype name you configured for this external business function press F1 and then enter. This will display a list of all the functions available for creating your expressions.

What happens if you enter an incorrect prototype, i.e. a prototype with a different number of parameters, different type of parameters, or with a different type of data in the result?

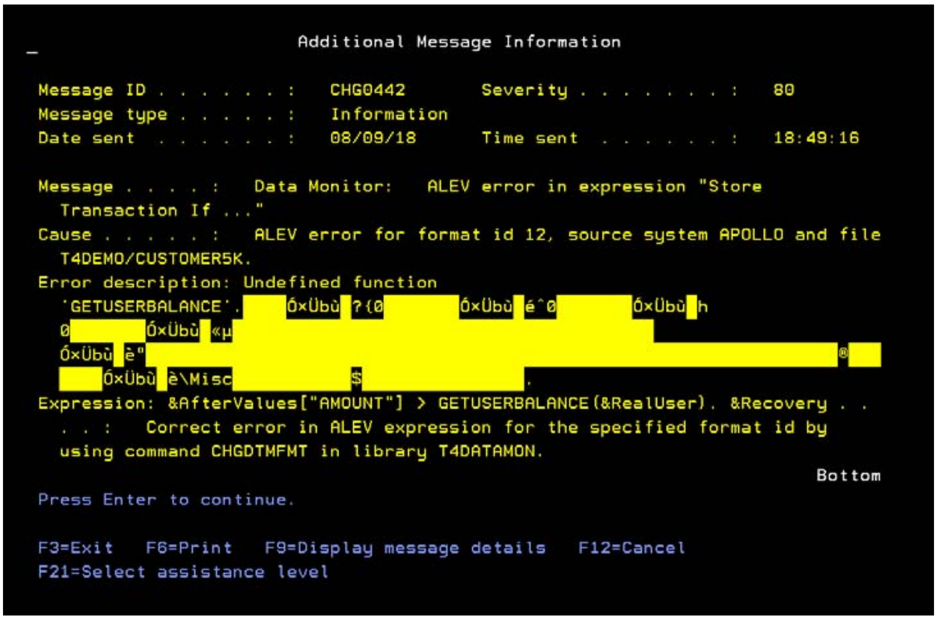
As far as Data Monitor is concerned this is a different prototype. If by chance a prototype with the specified parameters exists, you will receive unexpected results, as the calculations are made on incorrect data. If no such prototype exists the Data Monitor will give a syntax error.

Are the functions we use in our expressions case sensitive?

The functions are not case sensitive. However, we recommend the user write the prototype names in capital letters. You are then free to work with these prototypes in your expressions in whatever case you prefer.

What happens if you change an expression containing a correct function to use an expired function?

An ALEV error message will appear in the product message queue. You an access the message queue from the monitored files interface by pressing F9. It is important that you use only valid functions in your expressions. If a function expires it should be removed from all your expressions or replaced by a valid one.



```
                       Additional Message Information
_
 Message ID . . . . . . :   CHG0442       Severity . . . . . . . :   80
 Message type . . . . . :   Information
 Date sent  . . . . . . :   08/09/18      Time sent  . . . . . . :   18:49:16

 Message . . . . :   Data Monitor:   ALEV error in expression "Store
   Transaction If ..."
 Cause . . . . . :   ALEV error for format id 12, source system APOLLO and file
   T4DEMO/CUSTOMER5K.
 Error description: Undefined function
   'GETUSERBALANCE'.     ÓxÜbù ?{0       ÓxÜbù é^0       ÓxÜbù h
   0       ÓxÜbù «µ
   ÓxÜbù è"                                                         ®
       ÓxÜbù è\Misc           $                .
 Expression: &AfterValues["AMOUNT"] > GETUSERBALANCE(&RealUser). &Recovery . .
   . . :   Correct error in ALEV expression for the specified format id by
   using command CHGDTMFMT in library T4DATAMON.
                                                                        Bottom
 Press Enter to continue.

 F3=Exit   F6=Print   F9=Display message details   F12=Cancel
 F21=Select assistance level
```

Figure 91 – ALEV error message

What happens if you add new functions while your monitors are running?

There are three scenarios:

- You only add the new external business functions to Data Monitor but do not use them in your expressions. Data Monitor will continue to work properly without these new functions.
- You add new external business functions and use them in an expression. If there are no events related to these expressions with the new functions you defined, Data Monitor will work properly because these changes don't affect it.
- You add new external business functions, you use them in your expressions and there are events related to these expressions. In this scenario Data Monitor will find new undefined functions and an error will be sent to the product message queue. The situation here is similar to that explained in question 6. To solve this situation you must restart the monitors, so that they will be aware of the new functions.

What do I do if I want to make an expression with a new external business function and the Data Monitor expressions editor shows an “undefined function” error?

Most likely you have used the same interactive session to include the new functions and to create the new expressions using them. In order to work with these new functions in your expressions simple sign off and sign on again.

What happens if you delete or change an external business function defined in Data Monitor?

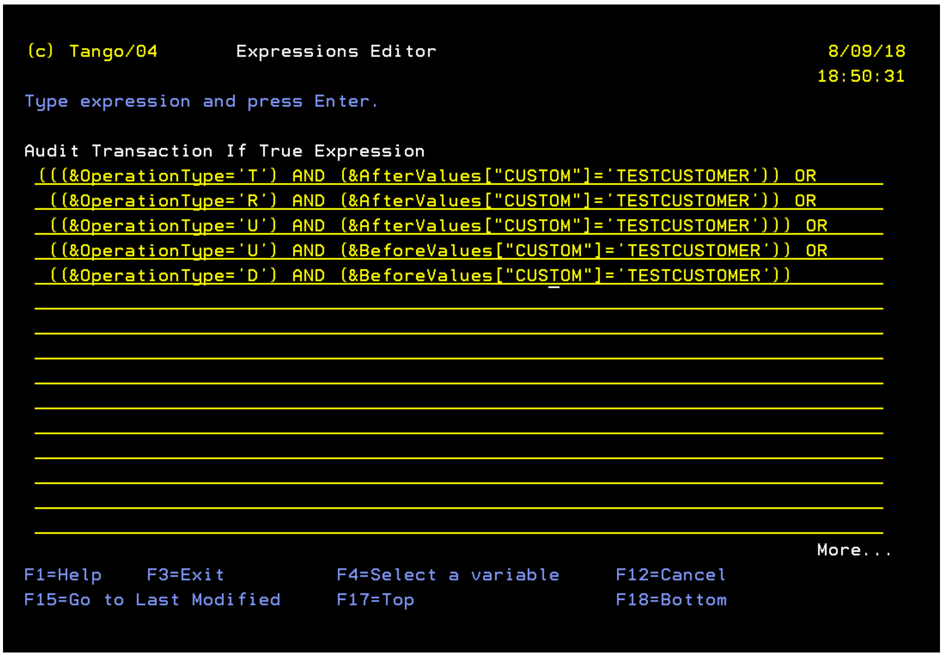
To avoid problems after deleting or changing an external business function defined in Data Monitor, it is important to check that they are also removed from all your current expressions.

If you restart a monitor containing an external business function that has been removed from `T4DATAMON/CHEFD01P`, Data Monitor will try to work with them and send errors to product message queue when it fails.

Appendix H
Using Advanced Filters and Expressions

H.1 Optimizing expressions with &OperationsValues

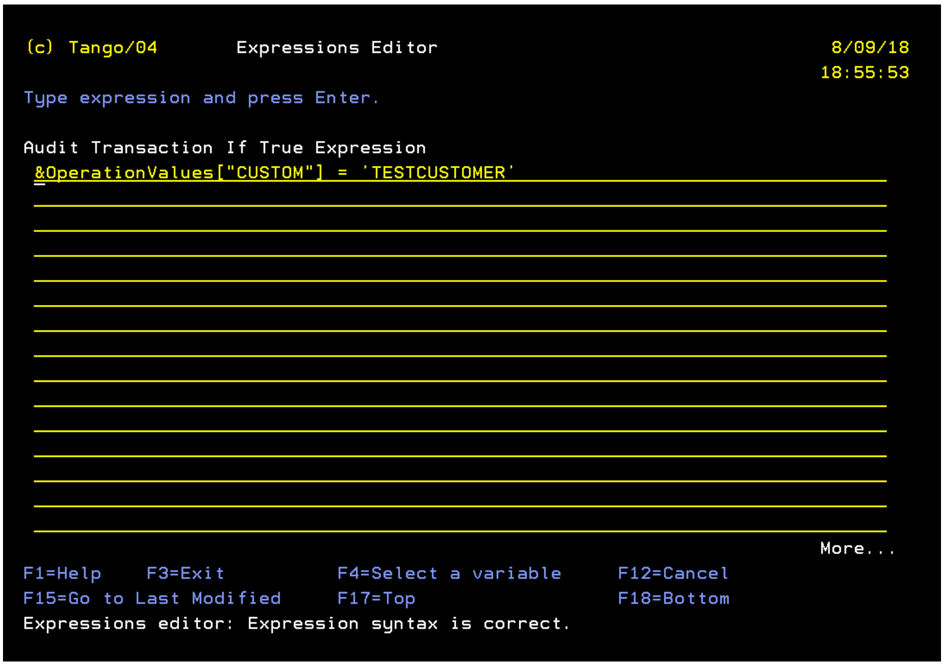
Imagine you want to audit only transactions where the field `CUSTOM` has the value `TESTCUSTOMER`. The first figure shows you the expression required using `&BeforeValues` and `&AfterValues`.



```
 (c)  Tango/04        Expressions Editor                          8/09/18
                                                                 18:50:31
 Type expression and press Enter.

 Audit Transaction If True Expression
  (((&OperationType='T') AND (&AfterValues["CUSTOM"]='TESTCUSTOMER')) OR
   ((&OperationType='R') AND (&AfterValues["CUSTOM"]='TESTCUSTOMER')) OR
   ((&OperationType='U') AND (&AfterValues["CUSTOM"]='TESTCUSTOMER'))) OR
   ((&OperationType='U') AND (&BeforeValues["CUSTOM"]='TESTCUSTOMER')) OR
   ((&OperationType='D') AND (&BeforeValues["CUSTOM"]='TESTCUSTOMER'))

                                                                More...
 F1=Help    F3=Exit          F4=Select a variable      F12=Cancel
 F15=Go to Last Modified     F17=Top                   F18=Bottom
```

Figure 92 – Expression required using &BeforeValues and &AfterValues

The second figure shows you how to use `&OperationValues` to simplify this expression.



```
(c) Tango/04       Expressions Editor                                   8/09/18
                                                                       18:55:53
Type expression and press Enter.

Audit Transaction If True Expression
 &OperationValues["CUSTOM"] = 'TESTCUSTOMER'















                                                                      More...
F1=Help    F3=Exit          F4=Select a variable     F12=Cancel
F15=Go to Last Modified     F17=Top                  F18=Bottom
Expressions editor: Expression syntax is correct.
```

Figure 93 – Use &OperationValues to simplify this expression

Note

For modifications (updates) - `&BeforeValues` stores values of fields before the record was updated and &AfterValues variable have values of fields after record update.

For inserts - variable `&AfterValue`s stores the value of the fields of the Added record.

For deletes - variable `&BeforeValues` stores the value of the fields of the Deleted record.

For reads - variable `&AfterValues` stores the value of the fields of the Read record.

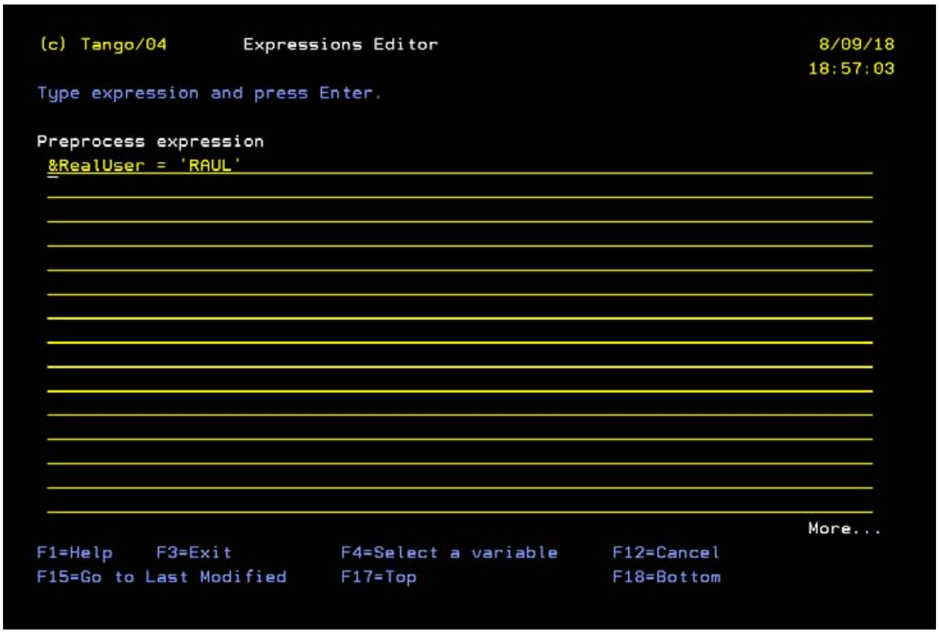
H.2 Example of a Format using expressions

Imagine a situation where you have a unique format for a file A. This file has a field called `MYREALUSER`.

You want to store transactions only when real user of the job doing the transaction is `PAUL`. You have the real user of the job in ALEV variable `&RealUser`.

Your configuration will look like this. From Work with formats:

Option **30** – Expression to store transaction…

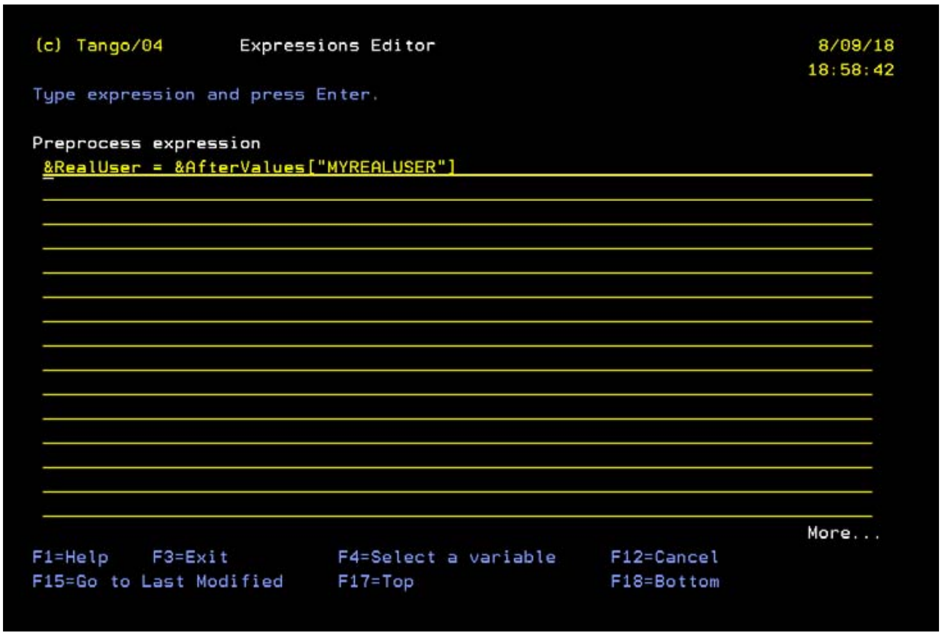


```
(c) Tango/04        Expressions Editor                              8/09/18
                                                                   18:57:03
Type expression and press Enter.

Preprocess expression
 &RealUser = 'RAUL'
                                                                    More...
F1=Help    F3=Exit           F4=Select a variable      F12=Cancel
F15=Go to Last Modified      F17=Top                   F18=Bottom
```

Figure 94 – Expression to store transacrions

In addition you want to store the value of your field `MYREALUSER` in the `&RealUser` variable.

Option **40** – Preprocess expression…



```
(c) Tango/04        Expressions Editor                              8/09/18
                                                                   18:58:42
Type expression and press Enter.

Preprocess expression
 &RealUser = &AfterValues["MYREALUSER"]
                                                                    More...
F1=Help    F3=Exit           F4=Select a variable      F12=Cancel
F15=Go to Last Modified      F17=Top                   F18=Bottom
```

Figure 95 – Preprocess expression

It is possible to introduce multiple expressions separated by semi-colons (;). For more information on ALEV, see the *ALEV 1.5 Reference Manual*.

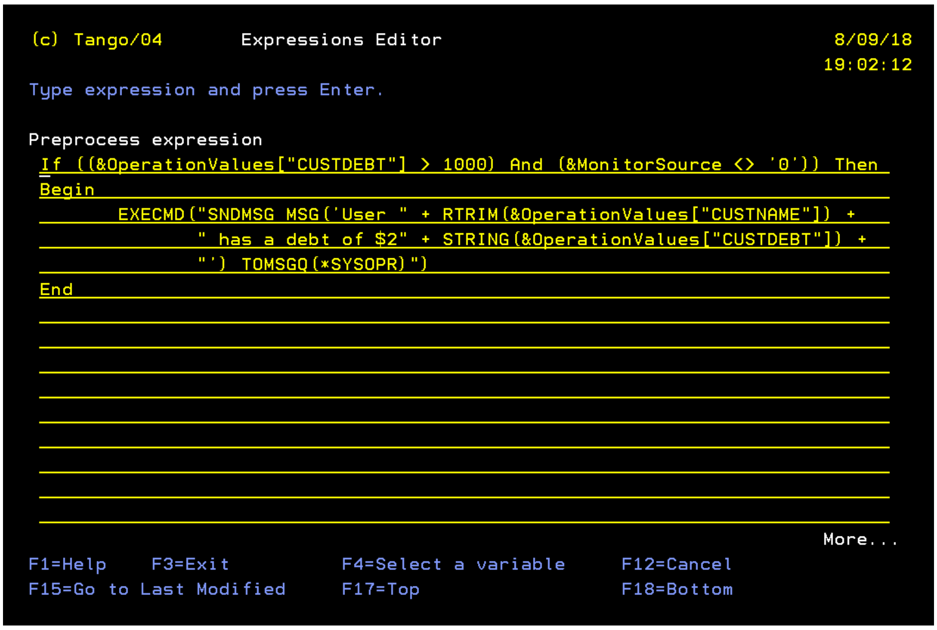
Note

When a user enters an expression in the expressions editor the expression syntax is automatically checked. If there are any errors in the expression a message will appear in the bottom of the screen and the user cannot exit the editor until a valid expression is entered.

Appendix I
Real Time Alerting for Important Transactions

I.1 Example 1

Imagine you want to be alerted as soon as possible when any customer has a debt greater than $1000. You can send an alert to any message queue with an ALEV preprocess expression. The next figure shows you the expression required, assuming that field CUSTNAME is the Customer Name and field CUSTDEBT is the current Customer debt:

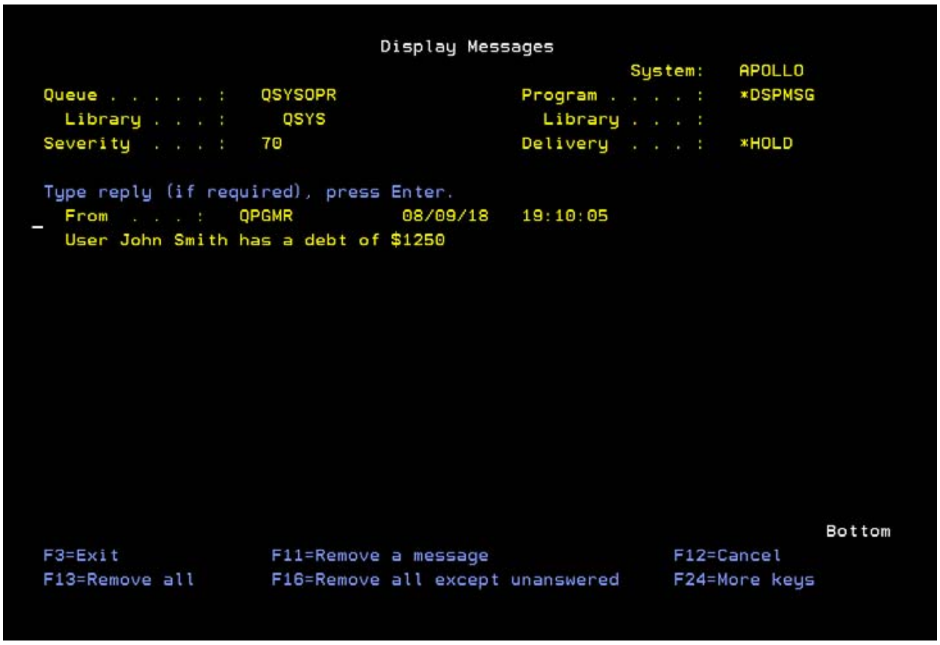


```
 (c)  Tango/04        Expressions Editor                                8/09/18
                                                                       19:02:12
 Type expression and press Enter.

 Preprocess expression
  If  ((&OperationValues["CUSTDEBT"] > 1000)  And  (&MonitorSource <> '0'))  Then
  Begin
         EXECMD("SNDMSG MSG('User  " + RTRIM(&OperationValues["CUSTNAME"])  +
                "  has a debt of $2" + STRING(&OperationValues["CUSTDEBT"])  +
                "')  TOMSGQ(*SYSOPR)")
  End










                                                                       More...
 F1=Help     F3=Exit          F4=Select a variable      F12=Cancel
 F15=Go to Last Modified      F17=Top                   F18=Bottom
```

Figure 96 – ALEV preprocess expression

The expression contains &MonitorSource <> 'O' because we do not want to be alerted when we are using Data Monitor for iSeries Express, which is not intended for Real time use. For a description of variable &MonitorSource or general ALEV information *see Appendix G: ALEV Variables and Functions for Expressions on page 109*.

With this expression you use function EXECMD to run command SNDMSG to send an alert to QSYSOPR message queue. The result of an update of customer debt to a value greater than $1000 should look like:



```
                                Display Messages
                                                       System:   APOLLO
 Queue . . . . . :   QSYSOPR                 Program . . . . :   *DSPMSG
   Library . . . :     QSYS                    Library . . . :
 Severity  . . . :   70                      Delivery  . . . :   *HOLD

 Type reply (if required), press Enter.
   From  . . . :   QPGMR          08/09/18   19:10:05
_  User John Smith has a debt of $1250




                                                                         Bottom
 F3=Exit              F11=Remove a message                  F12=Cancel
 F13=Remove all       F16=Remove all except unanswered      F24=More keys
```

Figure 97 – Displayed message

Besides sending an alert to a message queue you can also use any command in your system that provides you with a way of alerting in real time. For example send an email, write a record to another database table (with 360º functions).

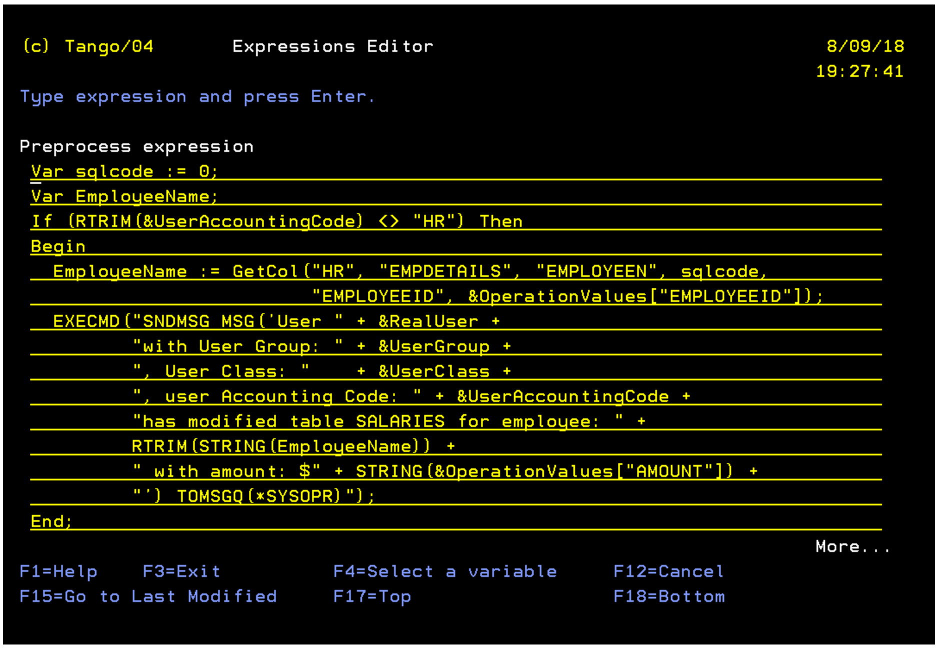
I.2 Example 2

We have a table called `SALARIES` which is usually only modified by users in the Human Resources department. So we want to be alerted if anyone outside this department changes this table.

We know (assumptions):

- All users in the Human Resources Department have accounting code `HR`.
- The `EMPLOYEEID` field contains the employee id, and the `AMOUNT` field contains the annual salary of the employee in the `SALARIES` table
- We have another table (not being monitored) called `EMPDETAILS` containing employee details. This table has a `EMPLOYEEID` key field and a `EMPLOYEEN` field which stores the employee name.

Using the following script for preprocess expression of the monitored table `SALARIES` we are alerted with a message in `QSYSOPR`.

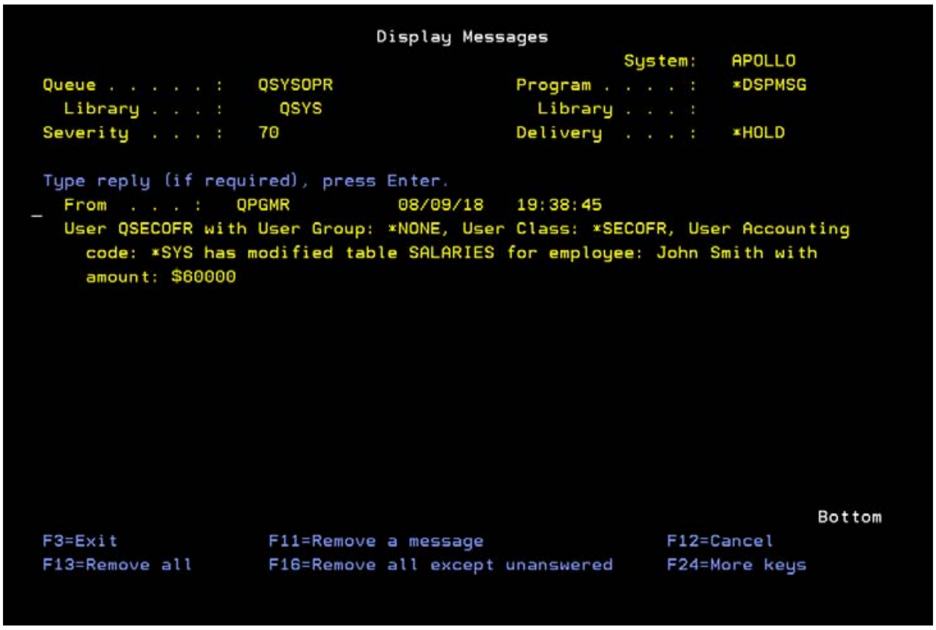


```
(c) Tango/04        Expressions Editor                                   8/09/18
                                                                        19:27:41
Type expression and press Enter.

Preprocess expression
 Var sqlcode := 0;
 Var EmployeeName;
 If (RTRIM(&UserAccountingCode) <> "HR") Then
 Begin
   EmployeeName := GetCol("HR", "EMPDETAILS", "EMPLOYEEN", sqlcode,
                        "EMPLOYEEID", &OperationValues["EMPLOYEEID"]);
   EXECMD("SNDMSG MSG('User " + &RealUser +
            "with User Group: " + &UserGroup +
            ", User Class: "    + &UserClass +
            ", user Accounting Code: " + &UserAccountingCode +
            "has modified table SALARIES for employee: " +
            RTRIM(STRING(EmployeeName)) +
            " with amount: $" + STRING(&OperationValues["AMOUNT"]) +
            "') TOMSGQ(*SYSOPR)");
 End;
                                                                       More...
F1=Help    F3=Exit          F4=Select a variable      F12=Cancel
F15=Go to Last Modified     F17=Top                   F18=Bottom
```

Figure 98 – preprocess expression of the monitored table SALARIES

If you look at the preprocess expression you can see that we first use the `GetCol` function to get the employee name (`EMPLOYEEN` field in `EMPDETAILS` table) by its `EMPLOYEEID` (employeed id). Then we use this employee name (stored in variable `EmployeeName`) to build the message we send to `QSYSOPR` message queue.

The Message will look like this:



```
                                Display Messages
                                                     System:   APOLLO
 Queue . . . . . . :   QSYSOPR                  Program . . . . :   *DPSMSG
   Library . . . :     QSYS                       Library . . . :
 Severity  . . . :     70                       Delivery  . . . :   *HOLD

 Type reply (if required), press Enter.
   From  . . . :     QPGMR          08/09/18   19:38:45
_  User QSECOFR with User Group: *NONE, User Class: *SECOFR, User Accounting
     code: *SYS has modified table SALARIES for employee: John Smith with
     amount: $60000









                                                                      Bottom
 F3=Exit             F11=Remove a message                 F12=Cancel
 F13=Remove all      F16=Remove all except unanswered     F24=More keys
```

Figure 99 – Displayed message

Note

Messages in any message queue can easily be received in Visual Message Center Smart Console, simply by using Visual Message Center iSeries Server Agent. So you can receive this message in `QSYSOPR` in your Smart Console to take advantage of its advanced monitoring, automation, visualization, and notification features.

For examples of other message queues *see section 11.5 - Integrating Data Monitor with VISUAL Message Center on page 89* or refer to the *Visual Message Center (iSeries Modules) User Guide*.

Appendix J
Data Monitor Commands

COMMAND	DESCRIPTION
ADDDTMF	Add Data Monitor File adds a file to be monitored Data Monitor.
ADDDTMU	Add Data Monitor User adds a user to be monitored Data Monitor.
ADDDTMFLD	Add Field of Format adds a field to an existing format for a file that is monitored by Data Monitor.
ADDDTMFMT	The command Add Format Definition adds a format to a file that is monitored by Data Monitor.
CHGDTAMON	The Data Monitor configuration command allows us to change the start configuration of Data Monitor and the data retrieval mode of files monitored in Data Monitor.
CHGDTUMON	The Data Monitor configuration command allows us to change the start configuration of Data Monitor and the data retrieval mode of Users monitored in Data Monitor.
CHGDTMF	The Change Data Monitor File command changes the Data Monitor configuration for a monitored file.
CHGDTMU	The Change Data Monitor User command changes the Data Monitor configuration for a monitored User.
CHGDTMFLD	The Change Field of Format command changes the field details of an existing format for a file that is monitored by Data Monitor.
CHGDTMFMT	The Change Format Definition command changes the format of a file monitored by Data Monitor
CPU	The Data Monitor - Changes Key command authorizes use of the Data Monitor – Changes for a specific CPU and until a pre-determined date. It also removes the date restriction once a full User License has been entered.
CPUREA	The Data Monitor - Reads Key command authorizes use of the Data Monitor - Reads for a specific CPU and until a pre-determined date. It also removes the date restriction once a full User License has been entered.
DLTDTMFLD	Delete Field of Format

© 2013 Tango/04 Computing Group

COMMAND	DESCRIPTION
DLTDTMLOG	The Delete Historical Records command deletes records from the transaction history of Data Monitor. You can filter the records to delete by using the command parameters.
DLTOVERFA	The command Delete Override File Attributes deletes the remote journal attribute of a file monitored by Data Monitor.
DLTRBKLOG	The Delete Rollbacked Transactions command deletes all records of rolled-back transactions in the transaction history of Data Monitor. Records to delete can be filtered by command parameters.
DSPDOMF	The Details of Monitored File command displays the file details and the associated monitor of a file monitored by Data Monitor.
DSPDTMF	The Display Data Monitor File command shows the file details of a file monitored by Data Monitor.
DSPDTMU	The Display Data Monitor User command shows the settings of a user monitored by Data Monitor.
DSPDTMFLD	The Display Field of Format command displays details of a field of an existing format.
DSPDTMFMT	The Display Format Definition command (DSPDTMFMT) displays the format of a file monitored by Data Monitor.
DSPDTMVER	The Display Data Monitor Version command shows the current version of Data Monitor on the system. This value represents the current version of Data Monitor installed on your system.
ENDDTAMON	The End Data Monitor command ends the jobs that monitor the files currently configured in Data Monitor.
ENDDTUMON	The End Data Monitor User command ends the jobs that monitor the users currently configured in Data Monitor.
GENJRNDTA	The Generate Journal Data command generates the data for Data Monitor for iSeries Express.
INSTALL	The Install Data Monitor command creates the appropriate working environment for Data Monitor to function. One of its functions is to secure all objects of the library with one single authorization list.
MGROLDLOC	In version 4.0 this command is used to migrate data from earlier versions. The Migrate Old Loc file command migrates the data from the historical transaction log files from Data Monitor for iSeries version 3.0 and earlier to the corresponding files for Data Monitor for iSeries 4.0 and up. For details see the Installation Guide.
OVRFILATT	The command Override File Attributes adds or changes the remote journal attribute of a file monitored by Data Monitor.
RMVDTMF	The Remove Data Monitor File command deletes a file monitored by Data Monitor. If the system name of the entry that you want to delete is not the same as the local system or the value of the ASP group name must not be ignored, press F9 to specify additional parameters.
RMVDTMU	The Remove Data Monitor User command deletes a user monitored by Data Monitor.
RMVDTMFMT	The Remove Format Definition command deletes the format of a file monitored by Data Monitor.
STRDTAMON	The Start Data Monitor command starts all jobs that monitor the files currently configured in Data Monitor.

COMMAND	DESCRIPTION
STRDTUMON	The Start Data Monitor User command starts all jobs that monitor the users currently configured in Data Monitor.
UNINSTALL	The Uninstall Data Monitor command removes the product Data Monitor from your system. This includes deleting the product library and all other objects that may have been created during or after the installation process.
WRKCFGDTU	The Work with Configuration Data Monitor Users command opens user monitor settings (F15 from Work with monitored users), where you can configure an out file, reuse file formats, list of journals, set time to journal.

Appendix K
Frequently Asked Questions

This chapter starts with general questions that apply to all three editions of Data Monitor for iSeries (i.e. Data Monitor for iSeries Sensitive Files, Sensitive Users, and Express. Next it handles questions regarding the individual editions.

K.1 Data Monitor for iSeries

I have a valid license key, and have been using the product for a while already. I've configured a monitor to audit reads and now none of my monitors work. The error message says my key is not valid. How is this possible?

There are two different product license Keys for Data Monitor: one for auditing changes and a second key for auditing reads. It is likely that you have been running Data Monitor with the first type of license key with monitors that audit changes only. Now that you have configured a monitor to audit reads, but do not have the reads license key, Data monitor does not start any of the monitors.

First reconfigure your monitor so that no reads are audited (set audit value for reads to *NO). Then you will be able to start the changes monitors as before. Then request the read key from your Tango/04 partner. Once you have both license keys installed you will be able to create monitors to audit all events and all monitors will run.

Note that the reverse is also true: if you had a key to audit reads and you created a monitor to audit changes, again none of the monitors would start and you would find the error message that the key is not valid. You must have both license keys installed to monitor all events with Data Monitor.

Note that the default setting for audit reads is *NO

When I add a member to a physical file monitored by Data Monitor, the following transactions are not retrieved. I must restart the monitor for the file to which I added the member for it to work. What happened?

You are using an iSeries version V5R1 or V5R2 and the API we use to read journals is limited this way. From version V5R3M0 onward, all future transactions are audited automatically without user intervention.

How much disk space will Data Monitor for iSeries use to store our transactions?

You can use this formula for every file you need to monitor to estimate disk usage:

```
Disk Usage (in KBytes):= N * (0.6  +  0.08*NF  + 2*(0.025*NF +RL))
```

Where:

- `N` := Number of stored transactions (Inserts/Adds/Update/Read)
- `NF` := Number of fields in file monitored by Data Monitor for iSeries
- `RL` := Record Length for file monitored by Data Monitor for iSeries (in Kbytes)

So, let's say we have a table A in library PROD with 10 fields and record length 2 kbytes. Then 10000 stored transactions will use:

```
10000 * (0.6 + 0.08*10 + 2*(0.025*10 + 2)) = 59000 Kbytes (about 58 Mbytes
of disk).
```

Note

Of course if you decide not to save all fields for every transaction this formula should be changed to take in account only stored fields (`NF`) and only the record length of these fields.

How many transactions is Data Monitor for iSeries able to process per second?

This depends on a number of variables. For example the percentage of transactions stored in our transaction log, the number of fields in the monitored file, and the record length of the files being monitored, among others. However we can make an approximation based on the following.

Every iSeries-i5 system has a CPW value, which is a performance benchmark. This value depends on the model and processor features. We can estimate that for every 4 units of CPW, Data Monitor can process and save one transaction per second using 100 % of CPU.

So, let's say we have an i5, model 520 with processor feature 7397. This means we have 2400 CPW. Then Data Monitor for iSeries will be able to process and store 600 transactions per second or about 2 million every hour using all system CPU, or 1 million per hour using 50% CPU.

Of course, this calculation is based on storing all transactions in Data Monitor transaction log. However you can dramatically increase performance by filtering the transactions to audit, using expressions in the Data Monitor configuration.

Note

Take into account that every monitor can only take as much as 1 processor of your system, so in multiprocessor systems every monitor will only be able to use 100/number of processors of your CPU.

Can I move Data Monitor’s database to another system or platform in order to reduce disk space on my iSeries?

Yes, a Data Monitor database that is located on an iSeries system can also be replicated to different platforms such as Oracle or SQL Server.

Leading database engines such as Microsoft SQL Server, Oracle, or IBM UDB DB2 include replication services, which can help you replicate the data in real time, near real time, or in bulk transfers. Additionally, there are some third-party tools that also provide replication services, and your company may be using one. You may also use simple SQL statements to copy the data, periodically sending the changes to the target database and deleting the source data on the iSeries system once copied.

Check with your database administrator for the destination database or refer to the destination database documentation to learn more about how to synchronize the data in the manner most convenient for you. Once it is replicated you can use Reports, which uses an ODBC connection, to retrieve data from Data Monitor’s replicated database.

Finally after replicating the data you can clean the database on the iSeries system. For details *see Chapter 7 - Cleaning Data Monitor files on page 61*.

My monitor is running constantly but there are not many transactions in the system. What could be causing the monitor to be in RUN status all the time?

Be careful you are not creating a loop of monitored events when using ALEV functions for 360º data enrichment. This can occur when you are working with databases monitored by Data Monitor for iSeries, when an ALEV function in one monitored database triggers other ALEV functions in other monitored databases, which then trigger the first ALEV function.

For example, if there is an ALEV function in `FILE1:  SELECT * FROM LIBRARY/FILE2`.

And an ALEV function in `FILE2: SELECT * FROM LIBRARY/FILE1`.

If both files are configured for monitoring read database operations, there is an infinite loop created, because whenever a transaction is made in the first file, it generates a transaction in the second file which again creates a transaction in the first and so forth.

K.2 Data Monitor for iSeries Sensitive Files monitoring

Why is my monitor not activated when I press F10?

To find the reason why a monitor was not started use F9 to open the message queue and find any error messages that may help determine the problem.

- One of the reasons may be that the user profile does not have authorizations for the Data Monitor job.
- It is possible that the file is not yet journaled or that the journal is inactive. Files must be journaled and the journals must be active for Data Monitor to work.
- Also check that the user profile has permissions for the file to be monitored by Data Monitor.

Data Monitor says I don’t have authorization to access files and journals. What should I check?

Make sure the user profile running Data Monitor has authorization for the file you are trying to monitor and the related journal. Another way to change the user profile is to change the job descriptions of `DTMMONITOR` and `STRDTAMON` as described below.

I’m having problems running Data Monitor due to permissions settings. What authorizations should I check?

To use Data Monitor make sure that the user profile of the Data Monitor jobs (Job description `T4DATAMON/DTMMONITOR`) has access to the files added to Data Monitor.

This user profile must also have *ALL authority for the journal object and *USE authority for journal receiver objects.

The default user profile is `QPGMR`, but you can change it (in Job Description `T4DATAMON/DTMMONITOR` and `T4DATAMON/STRDTAMON`) to add authorization for the required files.

What does error message CPF7053 with code 2 mean?

If no data can be retrieved for the date range you entered, the error message `CPF7053` with code 2 will appear. It means that the receiver chain is broken after the Date/Time you selected to start the monitors. You should start in a Date/Time range when Data Monitor will not find any broken receiver chains.

I can't find details regarding job, program, or remote address. How can I get this information?

From version V5R2M0 onward you can maximize the utility of Data monitor by specifying the values `*JOB`, `*USR`, `*PGM`, `*PGMLIB` and `*RMTADR` in the parameter `FIXLENDTA` when creating or changing your journals. This will retrieve the information you are looking for. For example change your journal using the following command on source system:

```
CHGJRN      JRN(HRESOURCES/SALARYJRN)      JRNRCV(HRESOURCES/SALJRN0001)
MNGRCV(*SAME)     FIXLENDTA(*JOB     *USR     *PGM     *PGMLIB     *RMTADR)
```

Where for `FIXLENDATA`:

- Specify `*JOB`, `*USR` to determine the job name and job user name of the transaction.
- Specify `*PGM`, `*PGMLIB` to determine the program and program library executing the transaction.
- Specify `*RMTADR` to determine the remote IP address where the transaction took place.

Why are the formats not behaving as I expected?

If two formats have the same Source System Name, File Name, File Library, ASPGroup value and Timestamp From value, the format will be selected in the order in which they appear in the work with formats screen.

I have changed, or plan to change, one of my files (for example adding a field, deleting a field or changing a field attribute). Do I need to make any changes to my Data Monitor configuration?

Yes, you must add a new format for this file in Data Monitor Configuration. For a detailed explanation *see section 3.4.1 - Adding a Format to a File on page 19*. It is important to understand that the "Use format from (Date and Time)" attributes for the new format must lie between the last transaction of this file with the old format and the date and time when file was changed.

The easiest way to make the changes is as follows:

1. Stop using the file
2. Stop Data Monitor for this file
3. Make the required changes to the file
4. Add the new format for this file to the Data Monitor configuration
5. Start Data Monitor for this file and the new transactions will be audited correctly

If you do not add the new format before using the changed file, you should add the new format to Data Monitor as explained above and then *see Appendix C: How to Re-Process Entries on page 99* to audit these transactions with new format in Data Monitor.

How many journaled files can Data Monitor for iSeries monitor in one journal?

Data Monitor for iSeries can monitor as many as 300 journaled files a single journal.

K.3 Data Monitor for iSeries Sensitive User monitoring

Why is my monitor not activated when I press F10?

To find the reason why a monitor was not started use F9 to open the message queue and find any error messages that may help determine the problem.

One of the reasons may be that the user profile of the Data Monitor job does not have the required authorization.

Also, make sure that you have entered the correct product activation key.

Data Monitor User says I don’t have authorization to access new journals. What should I check?

Make sure the user profile running Data Monitor has authorization for any of the journals you are trying to process. A way to change the user profile is to change the job descriptions of `DTUMONITOR` and `STRDTUMON` as described below.

This user profile must also have *ALL authority for the journal object and *USE authority for journal receiver objects.

The default user profile is `QPGMR`, but you can change it (in Job Description `T4DATAMON/DTUMONITOR` and `T4DATAMON/STRDTUMON`) to add authorization for the required files.

Do I need to define all the tables I want to monitor beforehand?

No. Data Monitor for iSeries Sensitive User can find transactions of any file in your system. It takes care of all the necessary steps for every file, such as for example loading a format when it finds the first journal entry for a specific file (Load Formats on the fly function). Where possible, we suggest you reuse the configured file formats (default setting in Data Monitor), so that Data Monitor will reuse your format configuration for sensitive files when monitoring sensitive users.

What does error message CPF7053 with code 2 mean?

If no data can be retrieved for the date range you entered, the error message `CPF7053` with code 2 will appear. It means that the receiver chain is broken after the Date/Time you selected to start the monitors. Data Monitor for iSeries Users may be working with more than one journal. In this case it will continue to monitor the remaining journals.

What does error message CPF709C mean?

To read transactions with Data Monitor for iSeries User the transactions must contain information about the real user. If the transaction does not contain real user information the message `CPF709C` will appear in the product message queue.

Make sure the parameter `FIXLENDTA` in your journal has been configured with the value `*USR`

From version V5R2M0 onward you can maximize the utility of Data monitor by specifying the values `*JOB`, `*USR`, `*PGM`, `*PGMLIB` and `*RMTADR` in the parameter `FIXLENDTA` when creating or changing your journals. This will retrieve the information you are looking for.

Why are the formats are not behaving as I expected?

If you are reusing formats in Data Monitor for iSeries Users, see the previous section for more information.

Data Monitor for iSeries Users automatically refreshes formats. In highly unusual circumstances it is possible that the formats are not synchronized with the timestamp of the transactions retrieved. In these cases Data Monitor will send the message `USR0473` including instructions on how to correct the situation.

When I checked the T4DATAMON subsystem I noticed the status of my DTUMONITOR jobs was hold, and later they were released again. Why?

`FMTCONTROL` is the job in charge of refreshing formats. In very specific situations you might find `FMTCONTROL` holds your jobs until the formats can be refreshed. Your jobs will be released as soon as the process is complete.

I have a correct product activation key, my monitor is active and running, the monitor was started with option *NOW, and there is a lot of information that should be monitored. However, the last historic date and time value does not reflect any changes, what is happening?

First check your expressions to make sure that the transactions are included in your monitor.

Furthermore take into account that performance depends on your configuration. One of the most important factors is the number of journals you are working with. It is possible that the monitor is busy reading other journals, and will get back the journal in due time. Remember that there are a number of parameters (F15) you can use to tune your monitors.

K.4 Data Monitor for iSeries Express

How can I activate Data Monitor for iSeries Express?

Data Monitor for iSeries Express can be activated by three different keys

- An individual product activation key for Data Monitor iSeries Express
- The VISUAL Message Center Security Suite key
- The Data Monitor for iSeries Base suite key

Data Monitor says I don’t have authorization to access files and journals. What should I check?

Make sure the user profile running Data Monitor has authorization for the file you are trying to monitor and the related journal. Data Monitor for iSeries Express runs in the same interactive job, if you use the command line. However, you can submit the `GENJRNDTA` command using a different effective user profile for this job.

I’m having problems running Data Monitor due to permissions settings. What authorizations should I check?

To use Data Monitor iSeries Express make sure that the user profile of the interactive job, or the effective user profile for the submitted job, has access to the files added to Data Monitor.

This user profile must also have *ALL authority for the journal object and *USE authority for journal receiver objects.

The default user profile is `QPGMR`. However, you can add authorization for the required files by changing the effective user profile for the interactive session or the submitted job.

What does error message CPF7053 with code 2 mean?

If no data can be retrieved for the date range you entered, the error message `CPF7053` with code 2 will appear. It means that the receiver chain is broken after the Date/Time you selected to start the monitors. Data monitor for iSeries Express may be working with more than one journal. In this case it will continue to monitor the remaining journals.

Why are the formats not behaving as I expected?

If you are reusing formats in Data Monitor for iSeries Express, see the previous section for more information.

Data Monitor for iSeries Express automatically refreshes formats. In highly unusual circumstances it is possible that the formats are not synchronized with the timestamp of the transactions retrieved. In these cases Data Monitor will send the message `OND0473` including instructions on how to correct the situation.

Appendix L
Best Practices

L.0.1 In general, try to avoid duplication of data.

For example, if a file is accessed by only one user and you are monitoring both the file and the user, data monitor will process the same transactions twice. If already monitoring the user for a different purpose, we recommend you do not repeat monitoring by file. On the other hand if you only need to monitor this file, you should not use data monitor for users, as it will monitor more data than you require.

Also, consider carefully whether to use `*ALL` or enter a specific user. When you use `*ALL` you include all users on your system, with obvious repercussions for performance. To reduce the amount of data processed by Data Monitor we recommend you configure only the specific users you really need to monitor.

Tip
Try not to repeat users. For example, if you choose to use user `*ALL`, do not add more users – this will only lead Data Monitor to monitor the same user twice.

L.0.2 Do not run Data Monitor for iSeries Express directly from the command line

When you open Data Monitor for iSeries Express from the menu, or you run the `GENJRNDTA` command from your command line, it runs in your interactive session. Depending on how much data you want to retrieve your interactive session might be locked for a significant period of time.

Try to avoid this situation by submitting the `GENJRNDTA` command from a different job, for example using the system command `SBMJOB`.

L.0.3 Use Key Description to enrich transaction information

You can use preprocess expressions to set the value of ALEV variable `&KeyDescription` with a more descriptive text of the key fields themselves.

Reports of these transactions will be easier to read by just looking at the Key Description. For example:

In your transaction you have a key field with 2 characters that store a product code. This product code is not an actual description of the product, but you have another table that stores a product description for each product code. You can use `GetCol` ALEV function to access this table and retrieve the product description for the transaction product code. This product description can easily be assigned to `&KeyDescription`. For more details and examples *see section 8.1 - ALEV functions used for 360º data enrichment on page 66*.

L.0.4 Do not monitor read transactions if it is not strictly necessary

The number of read transactions can be potentially huge. Therefore, it could take a great quantity of resources for Data Monitor to process them. In any case, try to reduce the monitor of read transactions to a minimum.

Appendix M
Contacting Tango/04

North America

Tango/04 North America
PO BOX 3301
NH 03458 Peterborough
USA

Phone: 1-800-304-6872 / 603-924-7391
Fax: 858-428-2864
sales@tango04.net
www.tango04.com

Italy

Tango/04 Italy
Viale Garibaldi 51/53
13100 Vercelli
Italy

Phone: +39 0161 56922
Fax: +39 0161 259277
info@tango04.it
www.tango04.it

Sales Office in Switzerland

Tango/04 Switzerland
18, Avenue Louis Casaï
CH-1209 Genève
Switzerland

Phone: +41 (0)22 747 7866
Fax: +41 (0)22 747 7999
contact@tango04.net
www.tango04.fr

EMEA

Tango/04 Computing Group S.L.
Avda. Meridiana 358, 5 A-B
08027 Barcelona
Spain

Phone: +34 93 274 0051
Fax: +34 93 345 1329
info@tango04.net
www.tango04.com

Sales Office in France

Tango/04 France
La Grande Arche
Paroi Nord 15ème étage
92044 Paris La Défense
France

Phone: +33 01 40 90 34 49
Fax: +33 01 40 90 31 01
contact@tango04.net
www.tango04.fr

Latin American Headquarters

Barcelona/04 Computing Group SRL (Argentina)
Avda. Federico Lacroze 2252, Piso 6
1426 Buenos Aires Capital Federal
Argentina

Phone: +54 11 4774-0112
Fax: +54 11 4773-9163
info@barcelona04.net
www.barcelona04.com

Sales Office in Peru

Barcelona/04 PERÚ
Centro Empresarial Real
Av. Víctor A. Belaúnde 147, Vía Principal 140
Edificio Real Seis, Piso 6
L 27 Lima
Perú

Phone: +51 1 211-2690
Fax: +51 1 211-2526
info@barcelona04.net
www.barcelona04.com

Sales Office in Chile

Barcelona/04 Chile
Nueva de Lyon 096 Oficina 702,
Providencia
Santiago
Chile

Phone: +56 2 234-0898
Fax: +56 2 2340865
info@barcelona04.net
www.barcelona04.com

Glossary

A

AccessServer

Tango/04's security application. It provides authentication and authorization services (like User Management and single sign-on) to all Tango/04 apps. Part of the Monitoring Engine (VMC-Mxx).

Action

A task that SmartConsole performs automatically when an Alarm is triggered. Actions include: writing to the Event Log, Executing an OS/400 command, Sending an e-mail, and more.

Action Set

A set of Actions that SmartConsole performs automatically when an Alarm is triggered.

Actor

An Actor, who plays an Application Role in Tango/04 apps, can be any Windows user or group, or another Application Role.

Advanced Assistance Services

A credit-based help system for Tango/04 Customers who need occasional help with their Monitoring Solution.

Agent Code (SmartConsole)

A ThinAgent's three-letter identification code. Each event arriving to SmartConsole from ThinkServer has this unique identification code, so you can: tell which ThinAgent the event came from; filter for events coming from a specific ThinAgent.

Agent Code (ThinkServer)

Represented by the variable `VSMAgentID`, this is a ThinAgent's three-letter identification code.

Agentless

Agents installed in ThinkServer—not in the target machine—are considered agentless because they do not involve installing on/consuming client resources.

Alarm

A SmartConsole filter that enables the user to be notified of certain events, perform actions, etc. An alarm may or may not include audible/visible alerts.

ALEV

iSeries Agents use a scripting language that allows the user to perform advanced operations or change the information contained in a message. This powerful yet simple scripting language is called ALEV and is explained in the Arithmetic-Logic Expression eValuator (ALEV)1.5 - Reference Manual. Part of the Monitoring Engine (VMC-Mxx).

Analysis Panel

The bottom panel of SmartConsole (Web Client) wherein the user can see the components being affected by a selected component experiencing a non-harmless health state. In the SmartConsole desktop client this panel is known as the Problem Root Cause panel.

Application Role

An Application Role is a user’s level of importance in a Tango/04 application. By default, three Application Roles exist: Viewer, Creator, and Administrator. New roles can also be created.

Applications Agent

A Legacy Monitoring Engine-standard product. Reads any real-time text file (from servers, applications etc.).

Auto-Enterprise View

The default way that SmartConsole Web Client displays the inner componenets of a selected Business Folder. It is for the sake of organization, as the user can see any subfolders and all the corresponding Business Views placed in their actual folders. An Auto-Enterprise View is not an Enterprise View.

B

Batch Queues Monitor

Included with VISUAL Message Center iSeries Base and the VISUAL Message Center Suite for Operations. Allows the monitoring of messages from any iSeries message queues. Not sold separately.

BEA WebLogic Operations Agent

A group of ThinAgents that monitor WebLogic Servers for bandwidth events and JMS messages exchanged between applications. Comes standard with the Monitoring Engine’s ThinkServer.

Browser Panel

The SmartConsole panel (at upper right) showing the components included in a selected Service Model Business Folder. As long as a folder is selected on the Service Model, Auto-Enterprise Views appear in this panel in SmartConsole Web Client.

BSM

See Business Services Management.

Business Category

A user-defined label for any SmartConsole Business Component. This property can be reused in VISUAL Message Center Reports and VISUAL Message Center Goals.

Business Folder

A folder anywhere on the SmartConsole Service Model. It can contain other Business Folders, Business Views, and SCPs. Business Folders are usually identifiable by a folder icon.

Business Network

The entire “Service Model” (or “Tree”) in the desktop SmartConsole interface.

Business Services Management (BSM)

A Monitoring Engine add-on that includes: nonstandard SLA Analytics and Reporting ThinAgents in ThinkServer; the ability to have SCPs and Total Impact Cost in SmartConsole; nonstandard BSM reports in Reports; special dashboards in Dashboards.

Business Value

A numeric weight given to a SmartConsole SCP, it represents the cost to your business that an event has on a monitored service or application. It is a user-defined number that should reflect lost profits, lost customers, or the like.

Business View

In VISUAL Message Center SmartConsole, a Business View represents an IT element of a business service or process. Business Views are the ends of the *branches* in the Service Model (*tree*). They group relevant incoming events and change Health status accordingly.

Business View Message Panel

In SmartConsole, the Browser panel becomes the Business View Message Panel when the user drills down, on the Service Model, to a Business View. The events (messages) corresponding to the Business View are listed in the panel.

C

Calendar

Created online via SharedObjects, a Calendar can be used across the range of Tango/04 apps to have an application consider only those days and hours set forth in the Calendar when handling events.

Child Reports

The number of reports that have been launched from a report template.

Cisco PIX/ASA Security Agent

These ThinAgents monitor your Cisco PIX and ASA firewall devices. This Agent does not come standard with the Monitoring Engine. Product code VMC-TSN.

Citrix Operations Agent

A sold-standard group of standard.ThinAgents that monitor Citrix farms and MetaFrame servers for processes, sessions, bandwidths, performance, datastore availability, users, and application load level, among other things.

Commander

The ThinkServer DOS-command line tool that enables users to quickly and easily list, start and stop monitors, and view events.

Connector

Determines where data used by Goals will come from. Connectors can be of type SmartConsole or ThinkServer.

Contact

A third party that is involved with a particular SLA (or goal) in VISUAL Message Center Goals. Contacts usually include: the name of the company, a primary contact person, and e-mail and phone details.

Contract

The binding SLA between you (your IT department or your organization) and another party, entered in VISUAL Message Center Goals. Contracts include: contact details for both parties, start and end dates, etc.

Crash Preventer

A product for the iSeries platform that is included with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base.

Criticalness

The health-state of a Business Component in SmartConsole: Harmless, Minor, Warning or Critical.

Criticalness Propagation

The way by which a Business Component in a Harmless, Minor, Warning or Critical state affects a parent component. User-defined, can be set to: Cause/Effect, Simple Redundancy, Custom Rules, etc.

Custom Command Agent

This ThinAgent group lets you execute single or multiple commands and scripts on remote Windows and Linux/UNIX systems, controlling the output returned. It has a wide range of use. A standard Agent.

D

Dashboards

VISUAL Message Center Dashboards is Tango/04’s real-time dashboard application. Including many predefined dashboards, it lets users view everything from system performance to business trends. The dashboards include charts, tables, gauges, and can handle SmartConsole branches, and can be shared across installations. Product code VMC-DSB.

DashboardServer

A legacy product, it was replaced by VISUAL Message Center Dashboards.

Data Adapter Agent

This ThinAgent group allows you to collect a wide variety of events for ThinkServer using SQL queries via ODBC connections. Pre- and post- queries are also used to modify these events to your liking. Monitoring Engine-standard.

Data Collector for Goals

A group of ThinAgents acting in ThinkServer to collect the data that Goals needs. This Agent is sold and used by Tango/04 Customers without a Monitoring Engine (a Customer having just Goals, for example). It has product code VMC-TOG.

Data Monitor for iSeries

Available for both Reads (DMI-REA) and Changes (DMI-BAS), and in an Express version as well (DMI-EXP), these products monitor IBM DB2 UDB databases, and comes with best-practices that are useful in security auditing. It is installed on the server you wish to monitor.

Database Settings Administrator

Tango/04's database management program for all Tango/04 apps. It lets you create each application's database, manage event signatures, variables, and define retention policies. Part of the Monitoring Engine (VMC-Mxx).

Debugging Agent (powered by VDW)

VISUAL Message Center Debugging Agent is an iSeries debugging product that enables a user to monitor and control Jobs, applications, Security, system information. It displays problematic source code and lets the user run commands and modify variables. Formerly known as VISUAL Debugger for WIndows. Product code VDW.

Dependencies

A relationship between one SmartConsole alarm and another. Specifically, if the first alarm is triggered, then the second one is automatically triggered as well.

Device Monitor

A product for the iSeries platform that is included with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base. It notifies the user of the statuses of drivers, devices, and communication lines.

E

Element ID

Used in ThinkServer—in Data Collection for Dashboards—it is a user-defined text variable that appears with an event (to distinguish it from others) when a monitor is producing multiple events per recollection.

Enterprise Problem Solver

A Legacy product, code EPS.

Enterprise View

Provides a top-down, high-level view in SmartConsole. Constructed using Service Model elements, you can add Business Views, Business Folders and SCPs to an Enterprise View, create custom backgrounds and place the selected elements exactly where you need to. You can even create links to other Enterprise Views or URLs, offering quick access to important information.

Escalation List

A list of e-mails/mobile phone numbers (for SMS) by which SmartConsole tries to contact operators when an alarm in triggered. SmartConsole tries first to contact the operator at the bottom of the list, and if no reply is received (or receipt of message delivered) within a certain amount of time, it moves up the list, attempting to contact the next operator, until an answer or appropriate response is received. The Escalation List is editable in Notifier, which handles the sending of the e-mails and SMS (text) messages for SmartConsole.

Events Integrity Agent

This Monitoring Engine-standard Agent (one ThinAgent, in this case) provides security to the *T4EventLob* table. It verifies the digital signatures of events and detects unauthorized modifications or deletions.

Exchange Server Operations Agent

There are two of these Agents: one for Exchange 2003, and the other for Exchange 2007-2010. They both include several ThinAgents that monitor the performance of the Exchange mail system itself as well as the relevant databases, services, DAGs, mailboxes, etc. Product code VMC-TOE.

Exclusion Range

A period of time in a Calendar that is not considered by a Tango/04 application in which the Calendar is used.

F

Filters

Used widely by VISUAL Message Center products to facilitate the easy retrieval of specific data.

Folder

See *Business Folder.*

G

Global Variables

ALEV variables that the user can define in SmartConsole, for use in scripts.

Goals

VISUAL Message Center Goals is a Tango/04 product that provides real-time calculation for SLAs and goals making sure you reach important service goals and avoid failures that can impact on your clients or can come from your suppliers.

H

Health

There are five possible health values that represent the current monitor status: *unknown*, *critical*, *warning*, *minor*, and *success*. The user can define monitor health conditions with the Health and Actions Wizard. When defining the health condition assignment you can use any of the variables available in the monitor. Every time the data source state is refreshed the configured health scripts are executed to assign the new monitor health.

Health Action (ThinkServer)

A Python script that a ThinkServer monitor runs when an event is found to match the conditions of a Health state.

HP Systems Insight Manager Agent

A group of ThinAgents that capture, from your current HP Systems Insight Manager software, the statuses and performances of your HP logical disks and controllers, memory modules, hardware fans and temperature, network interface adapters, physical disks and controllers, power supplies and consumptions, processors and CPUs. Standard with Monitoring Engine.

Hyperhashing

Tango/04 technology that boosts the efficiency of SmartConsole filters so that incoming events are processed up to 99% faster.

I

IBM Director SNMP Agent

This Agent is made up of one ThinAgent that lets you incorporate the physical and virtual IBM infrastructure currently monitored by IBM Director. Processing, disks and memory, configuration, and device-health events appear in ThinkServer. Comes standard with the Monitoring Engine; not sold separately.

Impact Analysis

SmartConsole's way of showing a user how one component is affecting another (how a Business View is affecting SCPs, for example).

Indicator

In VISUAL Message Center Goals, an Indicator performs the calculation (and/or Python expression) for each inflowing variable instance. An Indicator's output is compared against the conditions of a corresponding goal; this determines the goal's status.

Interactive Agent

A product for the iSeries platform that is only available as part of the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base. It monitors for any interactive error in the system and allows for automatic freezing of the user session. Not sold separately.

iSeries Agentless Security

Only available as part of the VISUAL Security Suite - Security Package (VSS) and the VMC iSeries Security Agent, these ThinAgents are partly agentless and return Security events based on job inactivity, system and network attributes, etc. Not sold separately.

iSeries Audit Journal

Available as part of either the VMC iSeries Security Agent or the VISUAL Security Suite - Security Package (VSS). Spawns events based on the iSeries Audit Journal. Not sold separately.

iSeries Operations ThinAgents

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, these ThinAgents are partly agentless and return Operations events based on queues, jobs, OUTFILE output, etc. Not sold separately.

iSeries Performance Agent

VISUAL Message Center iSeries Performance Agent also known as VCW, is included in the VISUAL Message Center Suite for Operations as well as VISUAL Control Center. Available as a standalone product. It monitors the performance of iSeries systems. Product code VCW-ISE.

iSeries Performance Manager

VISUAL Message Center iSeries Performance Manager is included in VISUAL Control Center, and is made up of two products: VISUAL Control for Jobs and VISUAL Control for Pools. It boosts system response time and automates job control and optimizes memory for pools. Product code VMC-VJP.

iSeries Security Agent

VISUAL Message Center iSeries Security Agent is also available as a standalone product, this Agent comes as part of the VISUAL Security Suite - Security Package (VSS). It is particularly useful for Security auditing and automated protection. Product code VMC-SEC.

iSeries SQL Agent

VISUAL Message Center iSeries SQL Agent includes the iSeries SQL Interactive Monitor and the iSeries SQL Monitor. It is useful in securing iSeries SQL databases and increasing application availability. Product code VMC-SQL.

iSeries Support Agent

VISUAL Message Center iSeries Support Agent also known as VISUAL Support, is included in the VISUAL Message Center Suite for Operations as well as VISUAL Control Center. It has a Remote Control functionality and is especially beneficial to programmers/technicians in situations where applications are undergoing big changes. Product code VMC-RST

iSeries Restricted State Monitor

VISUAL Message Center iSeries Restricted State Monitor requires either the Monitoring Engine or VISUAL Message Center iSeries Base. It delivers informational events when an iSeries system goes into a restricted state. Product code VSP.

J

JBoss Operations Agent

A ThinAgent group that comes standard with the Monitoring Engine (not sold apart). They monitor JBoss Application Servers and produce events related to memory pools, threads, CPU, and message queues.

JMX Generic Agent

This (single) ThinAgent is shipped with the Monitoring Engine, and is not sold individually. It monitors target MBeans and their attributes for Java application resource usage, response times and server performance.

Job Activity Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, this ThinAgent is useful in reporting unfinished or prematurely finished iSeries jobs. Not sold separately.

Job Duration Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, this ThinAgent produces events when an iSeries job exceeds predefined times. Not sold separately.

Job Status Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, this ThinAgent produces events if an active job has spent too long in a given abnormal state, such as MSGW or INEL. Not sold separately.

Job Que Status Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, this ThinAgent monitors job queues for the presence of problems such as full or blocked queues, jobs in excessive HLD or RLS, etc. Not sold separately.

K

KeyInfo

An XML file that you create with NiceLink Configurator. It contains the serial number and MAC address of a system for which you need a product activation key.

L

Label

A user-defined piece of text that appears on a dashboard.

Launch

To run a report based on a template in VISUAL Message Center Reports.

Link Generator

A VISUAL Message Center Dashboards dashboard sharing tool. A dashboard shared this way can be used—with full functionality, if desired—by another Dashboards user..

Linux Operations Agent

A nonstandard ThinAgent group for Linux and FreeBSD Operations. They spawn events related to physical and virtual memory, CPU, processes, file systems, etc. Product code VMC-TOL.

Linux Security Agent

A non-standard ThinAgent group for Linux and FreeBSD Security. They monitor, via SSH or Syslog, user and group activity, file systems, audit logs, etc. Product code VMC-TSL.

M

Managed Administration Services

A service offered by Tango/04 whereby a Customer’s Monitoring Solution is managed remotely.

Message Panel

Also known as the Message Grid, this is the upper panel in SmartConsole that shows the events filtering in to a Business View.

Monitor

Any instance of a ThinAgent. For example, to monitor the Availability of two different computers on your network you create two different monitors from the Network Basic Agent’s Ping ThinAgent—one for each IP address.

Monitoring Engine

Tango/04’s standard monitoring solution. It is composed of ThinkServer (with its various standard ThinAgents), SmartConsole (and its Web Client), Reports, and also the following middleware: AccessServer, Database Settings Administrator, Notifier, NiceLink, SharedObjects, and Scheduler. It also includes two Web Portal Users. Product code VMC-Mxx.

Multiple Copy

An action on any part or parts of the SmartConsole Service Model or Enterprise Views, it involves the duplication of the copied component(s) while renaming the pasted component(s). You can save a Multiple Copy for use later, by exporting it (this is only possible in the desktop console).

N

Narrow Table

The type of table in VISUAL Message Center Dashboards that typically includes one host, several variables (aligned vertically).

Network Basic Agent

A Monitoring Engine-standard ThinAgent group that monitors ports, POP3, SMTP, and performs pings to check network devices.

Network Operations Agent

This ThinAgent group does not come standard with the Monitoring Engine. It monitors Cisco Operations (memory, processors, temperature, interfaces, top flows). Product code VMC-TON.

Network Syslog Agent

A non-standard ThinAgent that captures events from the Syslog Daemon for use in ThinkServer (and thus, all other VISUAL Message Center apps). Product code VMC-TNW.

NiceLink

A network protocol created by Tango/04 to manage different communication protocols, such as TCP/IP, PPC and SNA Server, allowing VISUAL Message Center SmartConsole (and other Tango/04 products) to communicate with servers running different operating systems and different network protocols. Part of the Monitoring Engine (VMC-Mxx).

Non Resilient Mode

A SmartConsole operation mode that disregards events sent to the console at times when it is off. Faster than the resilient mode, especially in cases where the console has been down a long time.

Notifier

Notifier allows VISUAL Message Center SmartConsole (as an Alarm Action, for instance) to send SMS messages or e-mail from a mobile phone connected to a PC (the “server” phone) or an SMTP mail server to another mobile device (the “receiver” phone) or e-mail address, and reply to a message from any device via the server phone or via e-mail.

O

Operations Knowledge Module for IBM i

This product can be added on to: the Monitoring Engine with the VISUAL Message Center Suite for Operations, or the Monitoring Engine with VISUAL Message Center iSeries Base. System i features and functions in ThinkServer, SmartConsole, Reports and Dashboards. Product code VMC-KIO.

Oracle Operations Agent

Includes a wealth of ThinAgents; for Availability, Performance, and Alert Log, to name a few. Product code VMC-TOO.

Oracle Security Agent

ThinAgents that return Profile-, Role-, Login- and User-Manipulation events. Product code VMC-CSO.

P

Paste as Link

An action in SmartConsole whereby you are able to paste any Service Model component(s) onto another as a link. A pasted Business Folder icon appears with an arrow on it to indicate that is it a link, whereas the icon of a pasted Business View does not.

Player

A SmartConsole Web Client feature. Show a rolling, live presentation of desired Enterprise Views. Can be shared.

PMDB

Used by VISUAL Message Center to manage data source configuration and retention policies.

Policy Compliance for Windows

An Agent including several ThinAgents. They notify you when Windows Policies are changed, etc. Useful in Auditing for Regulatory Compliance. Product code VMC-TWP.

Portal User

A Tango/04 Portal User is an AccessServer-registered user for all desktop and Web applications. Two Tango/04 Portal Users are included with the Monitoring Engine. Extra Portal Users are sold in packages of Users under product code VMC-USE.

Post Event Group Action

An action that a ThinkServer monitor performs for an event group after Health rules are applied.

Post Health Check Action

An action that a ThinkServer monitor performs for each incoming event after Health rules are applied.

Pre Event Group Action

An action that a ThinkServer monitor performs for an event group before Health rules are applied.

Pre Health Check Action

An action that a ThinkServer monitor performs for each incoming event before Health rules are applied.

Preprocess

A combination of a filter that screens events and an ALEV expression that modifies them as soon as they reach SmartConsole (before appearing in the console).

Problem Root Cause Panel

In SmartConsole (the desktop client), this is the panel at bottom-right when any Business Folder is selected. It shows all the Business Views contained in the folder, their Total Impact, etc.

Python Generic Agent

Included with the Monitoring Engine, this ThinAgent can run any user-defined Python script at a specified time interval. Does not recollect data (used for scripting).

R

Range Set

A group of weekdays with desired hours (from a 24-hour scale) that you add to a Calendar. The days and hours set forth in a range set are the beginnings of a Calendar: the range set sets the days and hours that events will be considered.

Repetitions

SmartConsole alarm controls that determine if an alarm will repeat (or not), etc., if it has recently triggered.

ReportingSystem

A legacy product, replaced by VISUAL Message Center Reports.

Reports

A VISUAL Message Center module that allows you to create a variety of reports in real time or report on historical data. It uses ODBC to connect to data stored in the VISUAL Message Center databases and creates reports that allow you to view historical message information, Security, Auditing policies, User Usage/Accounting data, Performance Data, SLAs and more. This Web-based application is shipped with the Monitoring Engine.

Resilient Mode

An operation mode that allows SmartConsole to recuperate events that have occurred while the console was off or disconnected.

Restricted State Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, it produces events based on (`QSYSOPR`) messages when the iSeries system is in a restricted state. Not sold separately.

Role

See *Application Role*.

S

Scheduler

A piece of middleware that enables the execution of Tasks that you schedule in Dashboards and Reports.

Scorecard

A page within VISUAL Message Center Goals where the user sees the statuses of all goals. Goals has real-time, historical, and summary scorecards. Each are filterable (they can show specific goals if desired).

SCP

See *Service Control Point*.

Security Model

The product-specific security configuration stored by AccessServer for each Tango/04 product. A Security Model is important because it holds all the user permissions you have configured in a product.

Selector

A selection field in dashboards whereby the user chooses which data he wants the dashboard to show when he clicks the Show Dashboard button.

Service Level Agreement (SLA)

An agreement between an organization and a service provider, including those services provided by IT within the organization. In VISUAL Message Center SLAs are monitored by SLA Analytics ThinAgents in ThinkServer. In Goals: any goal. In SmartConsole: an SLA can be represented by an SCP on the Service Model. In Reports, BSM Reporting based on SCPs in SmartConsole is essentially SLA Reporting.

Service Control Point (SCP)

A type of folder that is used in Business Services Management (BSM) projects in VISUAL Message Center SmartConsole. Service Control Point (SCP) is a BSM concept and represents any critical element of a business, such as an application, a service, a business process, and more. SCPs have Business Values.

Severity

SmartConsole determines the Severity of each event coming in from ThinkServer. Severity then translates into Criticalness via the rules the user sets forth (in SmartConsole) for the four states (of Criticalness): Harmless, Minor, Warning and Critical.

SharedObjects

A repository for common data used by several VISUAL Message Center products (like calendars). It includes the PMDB statistical database which provides multidimensional statistical data, centralized access to BSM databases and forecasting functions.

SLA

See *Service Level Agreement*.

SLA Analytics

A group of ThinAgents that enable historical and real-time SLA events and reporting based on your SLAs. Sold separately. Product code VMC-MSL.

SmartConsole

VISUAL Message Center's client console. It has a desktop interface as well as a Web interface (SmartConsole Web Client). SmartConsole correlates events from ThinkServer via NiceLink and can perform actions on them (according to filters), simply display them in the console, or manage them to show the status of your business in corresponding locations on the Service Model.

SmartConsole Branch

A group of SmartConsole components shown in a dashboard exactly as they appear in a SmartConsole Service Model. In Dashboards the user can assign a display title, hyperlinks, and more for it.

SmartConsole Component

Any item on the SmartConsole Service Model (a.k.a. tree). The types of components are: Basic Service Control Point, Service Control Point, Business Folder, Business View.

SmartConsole Web Client

The Web-based client for VISUAL Message Center SmartConsole.

SNMP Agent

A group of ThinAgents for Network and Printer health, TCP Connections, and more. Shipped with the Monitoring Engine.

Spool Files by Job Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, it produces events when an iSeries job is using more spool files than allowed. Not sold separately.

Spool Files by Output Queue Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, it produces events when there are more spool files per output queue than allowed. Not sold separately.

Spool Files by Size Monitor

Included (exclusively) with both the VISUAL Message Center Suite for Operations and VISUAL Message Center iSeries Base, it produces events when it detects larger spool files than allowed. Not sold separately.

SQL Server Operations Agent

Has ThinAgents for Availability (like the Logical Disks Usage ThinAgents) and Performance (like the CPU Abuse ThinAgents). Not included with the standard Monitoring Engine, this Product code VMC-TOS.

T

Tag

A categorization of a report in Reports . . . like *Security*, *ThinkServer*, or *BSM*. Reports can have several tags at once. You can search for reports by clicking on tags.

Task

A scheduled macro with a defined trigger(s) in Dashboards and Reports. Tasks appear in the Scheduled Items list.

ThinAgent

Classes of monitors installed on ThinkServer. Many ThinAgents come standard with VISUAL Message Center ThinkServer, and many are available as will differ depending on what libraries you have on your server at the time of installation. A few examples are the Port, Ping, and SMTP Connect.

ThinkServer

The server component of VISUAL Message Center ThinkServer. ThinkServer contains all the reusable logic and stores the configuration the user sets in the ThinkServer Configurator. Therefore the ThinkServer can run independently from the Configurator. The ThinkServer usually runs as a Windows Service on your machine.

ThinkServer Configurator

The graphical client component of VISUAL Message Center ThinkServer. Here you can configure monitors, connect to one or more ThinkServers on your system, and view messages received from the ThinkServer. Note: a ThinkServer Configurator can be connected to more than one ThinkServer, and one ThinkServer may be connected to more than one ThinkServer Configurator.

Total Impact Cost

The sum of the Business Values of affected SmartConsole components. Shows operators which problems are affecting the business most negatively.

Trigger

A user-defined time-based definition that tells Dashboards or Reports when to run a macro. A component of Scheduler.

U

Universal File Reader Agent

Includes ThinAgents for Files and Folders. Reads logs. Comes standard with the Monitoring Engine.

Universal Transaction Agent

For SAP, Citrix, ActiveX control-using browser apps, and more. Available as a Monitoring Engine add-on. Includes a script runner (on monitored system). Product code VMC-TUC.

UNIX Operations Agent

ThinAgents for AIX, HP-UX and Solaris file systems, processes, and system health. Sold separately. Product code VMC-TSU.

UNIX Security Agent

A group of ThinAgents sold to complement the Monitoring Engine. They supply ThinkServer with events based on User Activity and Management, Files Systems etc. Product code VMC-TSU.

User Activity Monitor

Included with all iSeries suites (not sold separately). Generates events if user inactivity breaches a threshold of time.

User Message Variables

User-defined SmartConsole variables. They are per-message variables (they can have different values depending on the event). These variables can be used with ALEV.

V

Variable Instance

A VISUAL Message Center Goals concept. A single occasion whereby a desired variable (and its value, at that particular time) is passed to a Connector.

VISUAL Control Center

Includes VMC iSeries Performance Manager, the VMC iSeries Performance Agent (a.k.a. VCW), VISUAL Support Pro, and VISUAL Control Performance Planner. No Monitoring Engine required. Product code VCC.

VISUAL Control for Jobs

A single product. No Monitoring Engine required. Product code VCJ.

VISUAL Control for Pools

A single product. No Monitoring Engine required. Product code VCP.

VISUAL Control LPAR Tuner

Legacy product, code LPT.

VISUAL Control Performance Planner - All LPAR E.

A single product. No Monitoring Engine required. However, either the VMC iSeries Performance Agent or the VISUAL Message Center Suite for Operations *is* required. This product is also known as Performance Navigator. Product code VPP-IFL.

VISUAL Debugger 5250

A source-level interactive debugger that serves as support for batch and remote iSeries jobs. Product code VD.

VISUAL Message Center for iSeries/IBM i

The VISUAL Message Center Monitoring Engine for iSeries. Product code VMC-Mxx.

VISUAL Message Center iSeries Base

Includes several iSeries agents. A Monitoring Engine add-on. Product code VMC-BAS.

VISUAL Message Center Suite for Operations

An iSeries Monitoring Suite. Includes VISUAL Message Center iSeries Base, the VMC iSeries Performance Agent (a.k.a. VCW), VISUAL Support Pro, and VMC iSeries Restricted State Monitor. Product code SUI-OPE.

VISUAL Remote Control

Requires VISUAL Support Pro, but not the Monitoring Engine. Product code VSP-VRC.

VISUAL Security Suite - Security Package (VSS)

Includes the VMC iSeries Security Agent, Data Monitor for iSeries Express, and the VMC iSeries SQL Agent. Product code SUI-SEC.

VISUAL Support Pro

Available only with the VISUAL Message Center Suite for Operations. Inspects details of iSeries jobs. Product code VSP.

VMware Operations Agent

A Monitoring Engine add-on, it has ThinAgents for Role Management, VMs, Hosts, Datastores, Logons/ Logoffs etc. Product code VMC-TSV.

VMware Security Agent

Sold separately. ThinAgents for Role Management, VMs, Hosts, Datastores, Logons/Logoffs etc. Product code VMC-TSV.

W

Web Availability and Response Time Agent

Monitors that tell you how long Web transactions are taking on Consumer sites, their Availabilities and Contents, etc. Web 2.0-friendly. An End-User Experience Agent. Product code VMC-TWS.

Web SmartConsole

A Legacy product. The former Web interface for VISUAL Message Center SmartConsole (SmartConsole Web Client being the current one).

WebSphere Application Server Operations Agent

Agents for both Version 5.x and Version 6.x.The Agents includes ThinAgents for Thread pools, sessions, Enterprise Beans, and the J2C Connection Pool, etc. It is not shipped standard with the Monitoring Engine. Product code VMC-TOW.

WebSphere MQ Operations Agent

Safeguards Web applications. Includes several ThinAgents that create events about MQ Channels, the Queue Manager, and message traffic, among others. Product code VMC-TOQ.

Wide Table

The type of table in VISUAL Message Center Dashboards that typically includes several hosts (aligned vertically, and several variables (aligned horizontally).

Windows Operations Agent

Allowing deep insight into Windows Server Operations (Availability, Performance, Compliance etc.), this large collection of ThinAgents, which is sold apart from the Monitoring Engine, also delivers events about Services, Terminal Services, and File Systems. Product code VMC-WAD.

Windows Performance Agent

VISUAL Message Center Windows Performance Agent is a standalone product with code VCW-WIN. Does not require Monitoring Engine.

Windows Security Agent

A collection of ThinAgents, this Agent provides events based on Security: User/Computer, File, Domain Policy/Trust, Active Directory, and Event Log events appear in SmartConsole Business Views and predefined reports can be generated in Reports.

WMI Generic Agent

A ThinkServer staple, the ThinAgent runs WMI queries against as many local/remote Windows systems as you desire and returns provides disk, service, process, system configuration and Event Log events.

X

XML Generic Agent

Included with the Monitoring Engine, this ThinAgent reads any XML log/file anywhere to deliver events concerning service applications.

Index

© 2013 Tango/04 Computing Group

© 2013 Tango/04 Computing Group

About Tango/04 Computing Group

Tango/04 Computing Group is one of the leading developers of systems management and automation software. Tango/04 software helps companies maintain the operating health of all their business processes, improve service levels, increase productivity, and reduce costs through intelligent management of their IT infrastructure.

Founded in 1991 in Barcelona, Spain, Tango/04 is an IBM Business Partner and a key member of IBM's Autonomic Computing initiative. Tango/04 has more than a thousand customers who are served by over 35 authorized Business Partners around the world.

Alliances

Partnerships

IBM Business Partner

IBM Autonomic Computing Business Partner

IBM PartnerWorld for Developers Advanced Membership

IBM ISV Advantage Agreement

IBM Early code release

IBM Direct Technical Liaison

Microsoft Developer Network

Microsoft Early Code Release

Awards

Legal Notice

The information in this document was created using certain specific equipment and environments, and it is limited in application to those specific hardware and software products and version and releases levels.

Any references in this document regarding Tango/04 Computing Group products, software or services do not mean that Tango/04 Computing Group intends to make these available in all countries in which Tango/04 Computing Group operates. Any reference to a Tango/04 Computing Group product, software, or service may be used. Any functionally equivalent product that does not infringe any of Tango/04 Computing Group's intellectual property rights may be used instead of the Tango/04 Computing Group product, software or service

Tango/04 Computing Group may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

The information contained in this document has not been submitted to any formal Tango/04 Computing Group test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility, and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Despite the fact that Tango/04 Computing Group could have reviewed each item for accurateness in a specific situation, there is no guarantee that the same or similar results will be obtained somewhere else. Customers attempting to adapt these techniques to their own environments do so at their own risk. Tango/04 Computing Group shall not be liable for any damages arising out of your use of the techniques depicted on this document, even if they have been advised of the possibility of such damages. This document could contain technical inaccuracies or typographical errors.

Any pointers in this publication to external web sites are provided for your convenience only and do not, in any manner, serve as an endorsement of these web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries: iSeries, iSeriese, iSeries, i5, DB2, e (logo)®Server IBM ®, Operating System/400, OS/400, i5/OS.

Microsoft, SQL Server, Windows, Windows NT, Windows XP and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries. UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group. Oracle is a registered trade mark of Oracle Corporation.

Other company, product, and service names may be trademarks or service marks of other companies.