FLUKE®

Metrology Software

MET/CAL®
5500/CAL
MET/TRACK®

Users Manual

Contact MET/SUPPORT™ if you Need Assistance

See Appendix A for important details about the MET/SUPPORT program. Registered users of Fluke Metrology Software can obtain support by telephone, FAX, or E-Mail.

Before calling, be sure to first check the manual and online help.

Have the Following Information Available:

- Product name, applicable licenses, version and serial numbers
- Manual title, part number and version
- Operating system and version
- Network information if applicable
- List of steps necessary to recreate the problem
- Contents of `autoexec.bat` and `config.sys` files if applicable
- Version of *Crystal Reports*

Dial Toll-Free in the USA: 1-800-825-7411
FAX: 1-425-356-5992
E-Mail: metsupport@tc.fluke.com

Refer to Appendix A for international support and additional information.

FLUKE®

Fluke Metrology Software

Version 6

MET/CAL

5500/CAL

MET/TRACK

Users Manual

PN 1275392
July 1999

Table of Contents

Appendices

Index

List of Tables

List of Figures

Chapter 1
Theory of Operation

Contents **Page**

Safety Information

Calibrating instruments may require dangerous voltages to be generated or measured by the calibration system. Both the operator and procedure writer must be aware of possible hazards. To avoid electrical shock and fire hazards, follow the instructions below.

⚠Warning

To avoid hiding the high voltage safety symbol:

Never use a screen saver when using the Run Time or Procedure Editor applications. These programs control the test instruments and can cause them to output dangerous voltage levels. The high voltage symbol indicates when dangerous voltage is present.

⚠Warning

To avoid electric shock:

- **Use interconnect wiring and adapters rated for the highest voltage and current levels carried.**
- **Never touch exposed conductive portions of signal interconnect wiring, even after instruments have been turned off.**
- **Only touch the insulated portion of a connector. First disconnect the interconnect wiring at the source, then at the measurement side.**
- **Always use insulated adjustment tools.**
- **If the calibration program is abnormally terminated verify that the front panels of source instruments indicate standby mode.**
- **When writing procedures, make sure that procedures minimize the time during which dangerous voltages are applied.**

Introduction

Whether you are already using Fluke Metrology software, or this is your first experience, please read this chapter!

For those already using MET/CAL, 5500/CAL, or MET/TRACK, your upgrade will be more efficient if you can first appreciate the reasoning behind changes made in recent revisions. For new customers, you can be sure you are starting out on the right foot if you take the time to understand the product philosophy. This chapter explains how the software operates – the reasons it has been designed as it has, and what capabilities are provided to assist in solving particular problems.

Beginning with Version 4.3, Fluke Metrology Software fully incorporates all previous products into a cohesive set of applications. After installing the MET/BASE package, you apply licenses to enable those applications you wish to run.

About This Manual

This is the *Fluke Metrology Software Users Guide*; its purpose is to assist users who have already installed the base package and one or more licenses. There is no installation information in this manual. It provides specific information about calibrating instruments and managing metrology assets.

This chapter provides a look into the design considerations and ideas behind the software. Fluke's long-standing experience with calibration and metrology management software has proven that a solid conceptual model of the product gives users many advantages. Those who develop a clear picture of how the product works are better equipped to answer many of their own questions once they begin working with it.

Immediately After Installation

If you have not yet installed either the base package or any of the licenses, or if you are still in the process of setting up your database, you should first refer to the *Installation and Upgrade Guide.*

This manual contains complete information for setting up your database, configuring the system or customizing the data entry forms.

If you have purchased MET/CAL or 5500/CAL licenses, install the software first. Then, use the *MET/CAL Getting Started Guide* as an introduction to operation.

Possible Terminology Issues

- The 5500/CAL product is a subset of MET/CAL. Occasionally, the term MET/CAL is used even though both products are covered by the text. Rather than repeatedly using the phrase "MET/CAL or 5500/CAL", when you see just the term MET/CAL, you can presume that 5500/CAL operates identically. When there are differences between MET/CAL and 5500/CAL operation, that difference is always explicitly stated.
- This manual is intended to be used for random access. Do not read it from cover to cover. Some terms are used in early chapters that you may not become familiar with until you have worked with the software for a while. As a general rule, use this manual to locate the area most likely to contain the information you need, then begin reading. If terms or concepts are unclear, research them and, after you are familiar with the terms and concepts, return to the manual.

  If you come across an unfamiliar term while reading this manual, many resources are available. Look up the term in the glossary appendix, the on-line help, the Sybase on-line manuals, the README file, and so on.

This Manual Describes Only Database Mode

The previous versions of MET/CAL and 5500/CAL operated in two major modes: Legacy and Database. If you already have MET/CAL or 5500/CAL running in **Legacy Mode**, you must upgrade to **Database Mode**.

This Users Guide has answers to many different kinds of questions about Fluke's metrology software. It addresses two different audiences: those who set up, administer, and maintain the system, and those who are mainly interested in using the software to perform calibrations or tracking.

Intended Product Usage

The software applications design includes capabilities and features that will save time and effort in calibration laboratories. Even small installations can generate large amounts of information, and the traceability requirements built into new quality standards are making software solutions a necessity for meeting such day-to-day requirements as the following:

- List of instruments due for calibration
- Traceability reports for instruments used in calibrating other instruments
- Calibration certificates for instruments used in an ISO-9000 compliant organization
- Calibration results reports and calibration summary reports as defined by ISO 17025.

Traceability

You may be required to provide proof of calibration traceability, for example, to maintain ISO-9000 certification. In a metrology context, this means that you must be able to demonstrate an unbroken chain of comparisons between a given asset and a certified standard, such as one maintained by a national standards body.

When you enter information about a calibration event into the database, you are providing information about one link in the traceability chain. When you run a traceability report, you are viewing the entire chain of comparisons in one direction or the other.

A forward traceability report provides information about a particular asset and a particular calibration event; it is a listing of all the standards that are part of the chain of standards used to calibrate the asset for the event. The standards that are used in the current calibration can be called the first tier. The standards that were used to calibrate *those* standards are the second tier, those used to calibrate the second tier are in the third tier, and so on. You select the depth of the chain when you run the report.

A reverse traceability report documents all the assets calibrated directly or indirectly by the asset during a specified period. Again, the instruments that the asset of interest calibrated are the second tier, those that tier two calibrated are in tier three, and so on. You can find more information about traceability reports in later chapters of this manual.

Software Components

Two major components make up the Fluke Metrology Software product offering:

1. The MET/BASE package. This part is the core product, but it is not functional until the second component is in place.
2. The second component consists of one or more licenses that enable particular applications. You can install the client software on as many workstations as you like, because the database keeps track of how many licenses can be started at any given time. This is referred to as a “floating license.”

   The licenses available are called MET/CAL, 5500/CAL, and MET/TRACK. These are described next.

Applications Available to the Licenses

The following table lists the names of the applications that are enabled when you apply the various licenses.

	Licenses		
Application Name	**MET/CAL**	**5500/CAL**	**MET/TRACK**
MET/CAL Run Time	✓		
5500/CAL Run Time		✓	
MET/CAL Procedure Editor	✓		
5500/CAL Procedure Editor		✓	
Manual Data Entry	✓	✓	✓
Read-Only Manual Entry	✓	✓	✓

This chapter introduces these applications (and others that are not subject to license management). Details are provided in the remainder of this manual.

The Fluke Metrology Software Applications

The software is designed to assist users in conforming with ISO 17025, ISO-9000, ISO/IEC Guide 25, ANSI/NCSL Z540-1 and other requirements. ISO and other standards bodies have defined a structured process flow for metrology, and adhering to that flow is the only way to ensure calibration traceability.

The software is designed to integrate into this environment, so it is important for administrators to read this chapter. Calibration technicians, who may be working with the various applications every day, can also benefit from an understanding of these processes and requirements. The process flow is illustrated later in this chapter.

The Calibration Software is provided as a set of Windows-based applications that automate the calibration process. This includes procedure execution, instrument control, results storage, report generation, and asset management.

Because of the automation, the software improves calibration consistency, decreases the number of errors, reduces the amount of time spent calibrating an instrument, and provides information storage and retrieval.

The Run Time and Procedure Editor applications allow you to control calibration equipment on the IEEE-488 bus or RS-232 serial port, to verify and adjust a Unit Under Test (UUT), to generate reports, and to perform closed-loop verification and adjustment for UUTs with IEEE-488 or serial interfaces.

The following table shows the icons for all applications that can be installed. Note that the icons are in two groups: those representing Fluke programs, and those

provided as third-party software. Applications that are subject to license management include a large check mark beside the description.

With the exception of *Crystal Reports*, all of these programs are members of the Fluke Metrology Software program group that the installation program creates on the administrator's workstation.

Table 1-1. The Fluke Metrology Software Tool Set: Fluke-Developed Applications

	**MET/CAL and 5500/CAL Run Time** As the primary MET/CAL and 5500/CAL application, the Run Time executes calibration procedures and saves the results. Procedure Writers and Operators can access this application.	✓
	**MET/CAL and 5500/CAL Editor** The Editor helps you create, edit, and compile MET/CAL or 5500/CAL procedures. Procedures are the programs that run when you are calibrating instruments.	✓
	**Database Setup** The Database Setup application provides the interface for customization and system configuration. Database Setup can only be run by the MT user.	
	**Manual Data Entry** Provides access to the database for manual entry of data. This is the major (or only) interface used by data entry personnel for adding assets to the inventory and for recording events. Provides the mechanism for viewing and browsing data and has utilities for working with reports: adding, editing, and running them.	✓
✓ *Check mark indicates applications that are subject to license management.*		

Table 1-2. The Fluke Metrology Software Tool Set: Third Party Software

	**Database server** The server is part of the Sybase SQL Anywhere product. The server must be running before any client (application program) can be started. The icon is only seen in a standalone installation, or on the administrator's workstation in network installations.
	**Network requester client.** The network requester client is part of Sybase SQL Anywhere package. The server must be running before you start the network requester on the client machine. Once the requester is running, several applications can use it to connect to the same or a different database server.
	**ODBC Administrator** The ODBC Administrator is part of Windows. This tool configures the data sources.
	**ISQL** Interactive SQL is a part of the Sybase SQL Anywhere package. ISQL is an interactive tool for issuing commands in the Structured Query Language.
	**Crystal Reports** Crystal Reports is the optional report writer program used for generating reports. This is the application that was used to create the Sample Reports. For compatibility with supplied Sample Reports and to use the Manual Entry Tool for adding new reports, Fluke recommends using Crystal Reports to create any additional reports that may be required.

MET/CAL License

When you apply a MET/CAL license, you have access to the entire suite of software programs that assist a metrology lab in calibrating all kinds of equipment and in conforming with various standards requirements, such as ISO-9000.

5500/CAL License

The 5500/CAL License provides a limited version of the capabilities of a MET/CAL License. It includes drivers only for the Fluke 5500A, 5520A, 5800A and 5820A, the Hewlett-Packard 34401A and 34420A, 3325B, and the Keithley 2000 and 2024; The 5500/CAL License also limits control to the serial interface.

The 5500/CAL software does not support other system instrument drivers (e.g., the Fluke 5700A driver), nor access to the IEEE-488 interface.

MET/TRACK License

A MET/TRACK License provides a subset of the capabilities of a MET/CAL License. However, the Manual Data Entry application is the same as those enabled by MET/CAL or 5500/CAL Licenses, permitting all applications to work together in assisting users to conform with standards requirements.

Calibration Process Flow

You can make effective use of Fluke Metrology software just as it comes to you, or you can personalize it to make it fit better into your own processes and work flow.

The application design is based on the typical process flow of an instrument as it arrives, undergoes calibration or maintenance, and finally is sent back out to the location where it is used. If your requirements differ from this typical model, you can use the tools provided to make changes.

New Asset

Each piece of equipment (called an *asset)* receives a unique identification number when you first enter it into the system. The asset number is used throughout the database to refer to the asset.

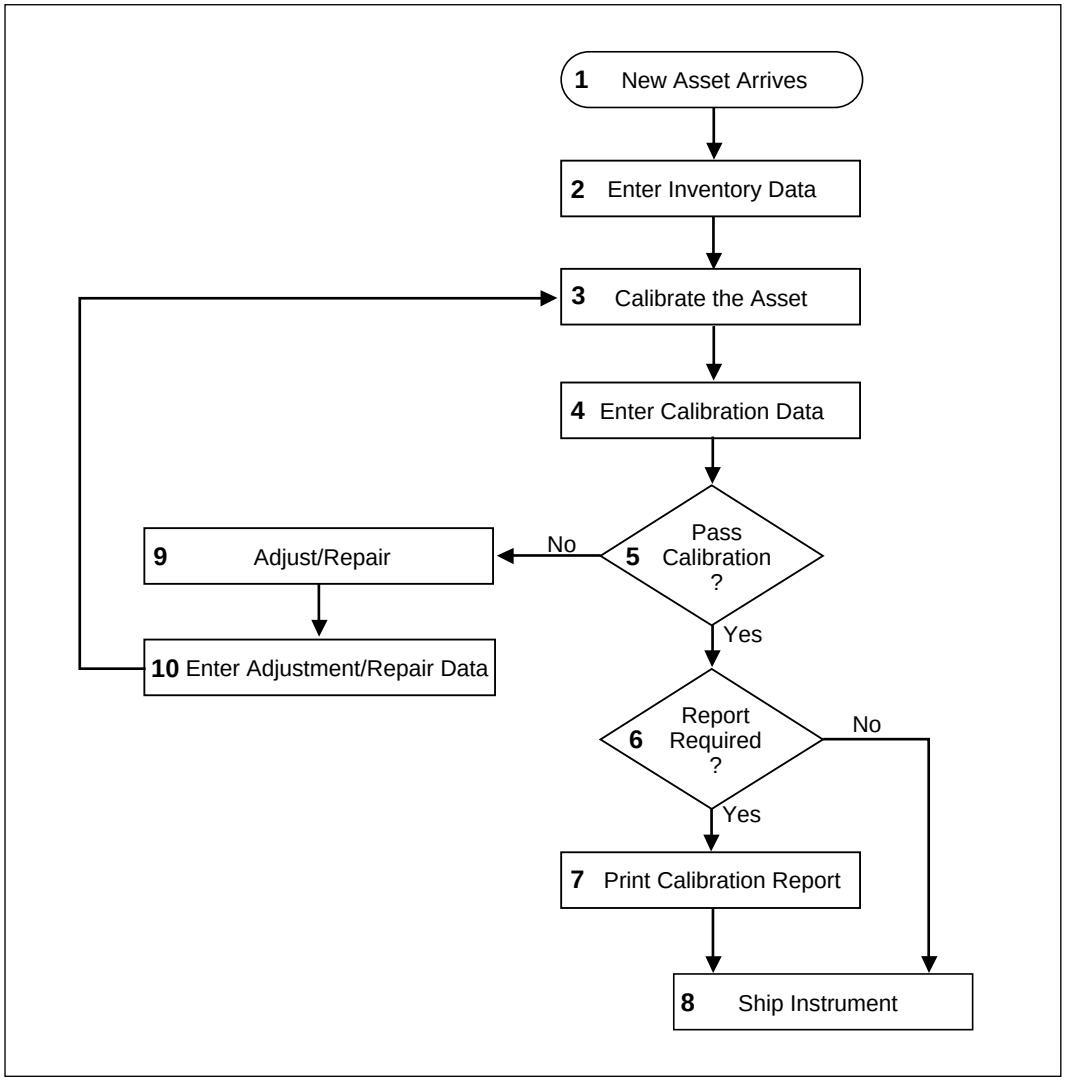


Figure 1-1. Process Flow 1: New Asset

Refer to the process flowchart in Figure 1-1 as you read the following description of the activities that normally take place in the calibration laboratory when a new asset arrives:

1 The new asset arrives on site.

2 Descriptive information about the asset is entered on an inventory form.

Using the Manual Entry or Run Time application, the inventory data is recorded in a database to identify the asset's calibration interval and arrival status, and to establish how access to the information will be controlled, depending on how the database is segregated.

An asset number is assigned at this time, uniquely identifying this instrument. To help maintain long term consistency of your data, Fluke advises against changing an asset's number once it has been assigned.

3 If necessary, the new asset is calibrated. Sometimes, an asset arrives with a recent calibration certificate.

Calibration can be done automatically using the Run Time application, or it can be done manually. If done with the Run Time, results are automatically recorded in the database.

4 A calibration record is written into the database. This can happen in either of two ways. First, if the asset is calibrated by a Run Time Calibration Procedure, the results are recorded automatically. The second way of saving the record is by way of the Manual Entry application, which would be used if you calibrate the instrument some way other than by using Fluke Metrology software.

If the asset was calibrated, the first record has the first calibration date and whether the asset passed or failed the initial calibration. Later on, you can retrieve this information for calculating new calibration due dates and for generating a reliability profile of the instrument.

5 If the asset passes calibration, it continues its incoming processing.

If not, it is adjusted or repaired (Step 9).

6 The need for a report is determined.

7 If a report is required, the Manual Entry or Runtime applications can print the Calibration Report.

8 In either case, the instrument leaves the calibration lab and is now ready to use.

9 At Step 5, if the asset did not pass calibration, it leaves the normal flow so it can be adjusted or repaired.

10 Repair or calibration results are entered into the database, and another attempt is made at calibration (back to Step 3). It has thus entered a loop that it will exit only when it passes calibration.

Asset Recalled: Due for Calibration

Refer to the process flowchart in Figure 1-2 as you read the following description of the activities that normally take place in the calibration laboratory when an asset is due for calibration. You will see that this process is very similar to the one followed when a new asset arrives, except that now the asset has been in use for a time. It may also have been used for calibrating other instruments, so a significantly out-of-calibration condition can have far-reaching consequences. You must test for this condition so you can notify owners of other instruments that may have been affected.

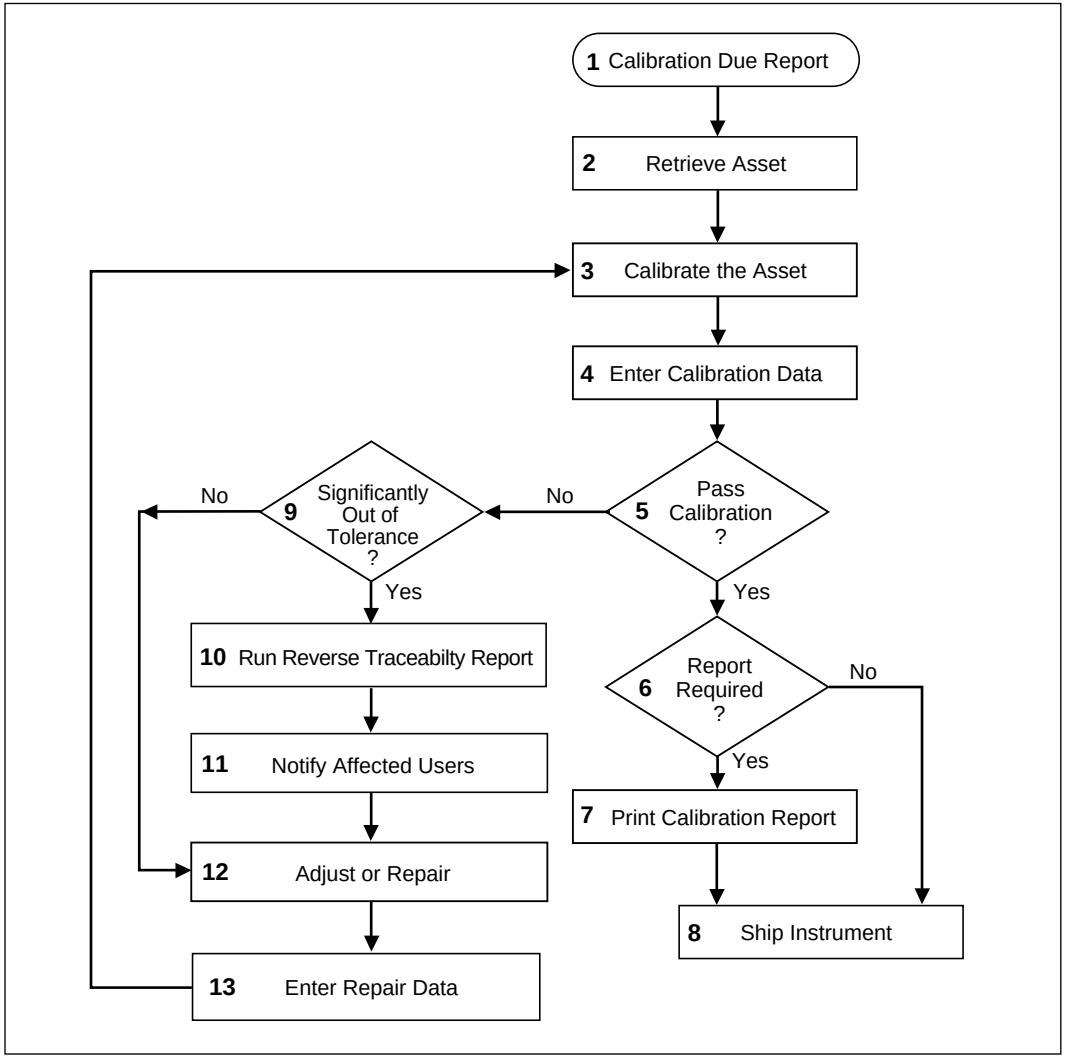


Figure 1-2. Process Flow 2: Asset Recalled, Due For Calibration

1 Periodically – daily, weekly, monthly, or perhaps once per quarter – a regularly scheduled *Due for Calibration Report* is run to identify assets that are due for calibration. Based on the results of running this report, an asset may be recalled for calibration.

As assets reach their calibration due dates, Fluke Metrology software helps to manage their flow through the lab. Calibration events occur when the calibration interval has elapsed.

2 The assets identified in the report are retrieved and brought to the lab.

3 The asset is calibrated, either automatically using the MET/CAL or 5500/CAL Run Time application, or manually.

4 Calibration data is entered into the database. If this was a manual calibration, then the Manual Entry application is used for entering the data. If the calibration was performed using one of the Run Time applications, the data is automatically entered in the database upon completion of the Calibration Procedure.

5 If the asset passes calibration, it is processed out just as it was when it was a new asset, through steps 6, 7, and 8.

If the asset did not pass calibration, the loop it enters takes a new factor into account: the tolerance level of the out of calibration condition. In effect, we must now ask, “How far out of tolerance is the Unit Under Test?”

6 If the instrument is found to be significantly out of tolerance, the assumption must be made that other instruments may have been calibrated by an out-of-tolerance instrument, since the safest premise is that the instrument has been out of tolerance since it was last calibrated.

Note carefully, however, that the definition of “significantly out of tolerance” can only be determined locally, based on the practices in place at each site.

7 A reverse traceability report is run to determine which other assets may have been affected by the out of tolerance instrument.

8 The owners of affected instruments are notified of the results so a decision can be made how to proceed.

9 The instrument is adjusted or repaired.

10 Adjustment or repair data is entered into the database and calibration is again attempted.

Asset Recalled: Due for Maintenance

If your system schedules periodic maintenance, you can use a Due For Maintenance Report to locate all the assets that are due for the maintenance. As assets reach their Maintenance Due dates, Fluke Metrology software helps manage their flow through your lab. Refer to the process flowchart in Figure 1-3.

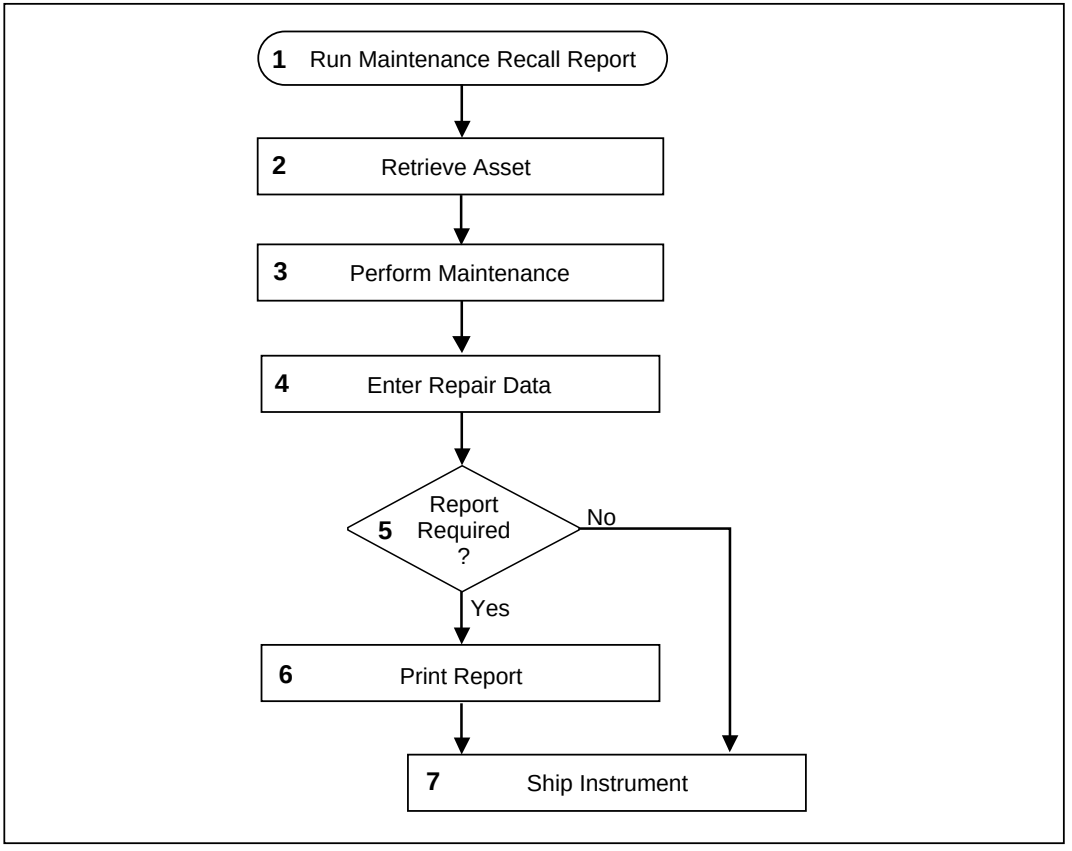


Figure 1-3. Process Flow 3: Asset Recalled, Due For Maintenance

The following paragraphs describe the activities that normally take place in the calibration laboratory when an asset is due for maintenance.

1 Periodically, the laboratory runs a report designed to locate all assets due for maintenance. This may be a part of a formal preventive maintenance program, or you may just like to bring in your instruments periodically for cleaning and inspection.

2 Based on the results of the Due for Maintenance Report, assets are recalled for maintenance. As assets reach their maintenance due dates, the Fluke

software helps manage their flow through the lab. Maintenance or repair events occur when the proper interval has elapsed. Queries create schedules while the Data Entry or Run Time applications record data about the assets, tracking basic asset information (Inventory), Calibration and Repair events, and where the asset is located.

3 The maintenance is performed.

4 Now the information about the maintenance that was performed is entered into the database. These records keep a complete history of everything that has ever happened to the asset. The record is begun as part of the asset's incoming processing, and added to every time there is a maintenance event. You can define even something as routine as cleaning a filter to be included in this record. These fields can be used to track PM histories, providing an opportunity to use calculated fields and generate a Due for Maintenance report if you like.

5 If you generate a results report regarding this maintenance, and one is required. . .

6 . . . use the Manual Entry application to print the report. This may be one of the sample reports that comes built into the system, or one that you have created yourself.

7 After the records are updated, the asset is returned to service.

Software Operation

MET/BASE uses two primary computer services:

- File Services
- Client/Server Network services

First, consider file services for a single user computer not on a network. Recall that Windows is a multi-tasking operating system, and many programs work together and execute (run) at the same time.

The executable program is stored on the hard disk. When the user selects a program to run, Windows sends a request to the disk manager to get the data that comprises the *exe* file. The disk manager calls a device driver that reads the program off of the hard disk, and returns the bytes to the operating system (Windows). Windows loads these into RAM and starts program execution. This is an example of a file service.

What happens when you run a program on a networked system? Essentially, the same series of events happens. However, in this case, the hard disk is not in the local computer (workstation). It is physically in a "file server", a computer located elsewhere. When the user selects a program to run, Windows still sends a request to the disk manager for the file. The disk manager still calls a device driver, but

this device driver is not for the hard disk, it is for a network interface. The network device driver sends messages to the file server. Programs running on the file server get these requests and use a device driver on the file server to read the *exe* program off of its hard disk. The *exe* data is sent back across the network, and Windows loads the program into RAM and starts program execution.

When MET/BASE is installed, it needs a “shared” directory, named METCAL by default. This directory is on the file server. In a default installation, ALL of the workstations on the network use the executable programs, reports, calibration procedures, and other files out of this shared directory. Note that for this to work the workstations involved are running several pieces of software (Windows, network drivers, and so on), and the file server must be running its software.

File services are fairly simple, and work very well in cases where the Client workstations read and write entire files. For example, if the MET/BASE editor loads a calibration procedure, it loads the entire procedure into memory, and no other user can change the procedure.

The Information Technology (IT) department typically sets up file services. These services are so common we typically don't even think about them. (Like driving a car but not fully understanding how the engine works.)

However, things are different with the database. It would be very inconvenient if only one user at a time could modify data, so the software makes use of Client/Server network services.

Client/Server Network Services

The following description traces the data from the database file back to an application on a networked system.

Refer to Figure 1-4 to trace the data from the database file back to an application on a networked system.

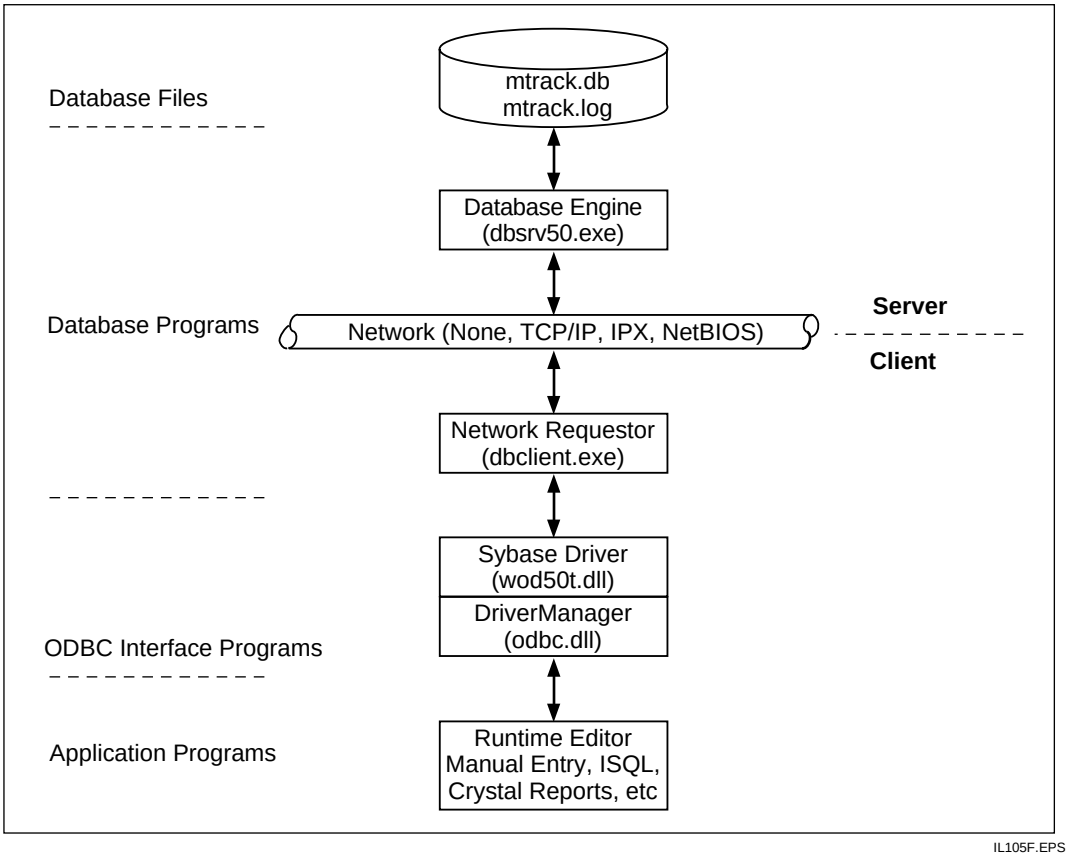


Figure 1-4. Diagram of Metrology Software System

For the system to operate, multiple software programs run simultaneously. Even on a standalone system, there are at least 5 programs running and working together at the same time.

The database file, **mtrack.db** resides on one of the computers on the network. Only the database engine (also called *data server* or *backend*) can read and write to this file. No other programs read, write or access the database file in any way when the database is in use. You can tell when the server is running, because you will see the server icon or window on the display.

This means that the data engine program must run on the machine that has access to the database file. That machine is referred to as the *database server,* the *data server*, or just the *server*, because the database engine is running on this machine, which may be running under Windows NT, Windows 95, or Windows 98. It is strongly recommended that only administrator tasks be performed on this machine.

So, the database engine is running on a computer. If the computer is on a network, it waits for database requests to come over the network cable. These requests must conform to one of the supported network protocols. Fluke Metrology Software supports TCP/IP, IPX, and NetBIOS network protocols.

Database requests come from client computers, which are also called workstations or nodes. A program called the *data requester* sends and receives messages to and from the data engine on the server using the network protocol.

On a standalone system, the same activities take place, including the requirement for requests to be in a supported network protocol. The main difference is that the data requester icon is not seen.

The client never directly reads, writes or accesses the database file in any way. You can tell the data requester is running if its icon is at the bottom of the Windows screen.

The messages for the data requester come from the client applications through ODBC (Open Database Connectivity standard). Client applications include the Query, Setup, and Manual Data Entry applications as well as the Run Time, Procedure Editor, Crystal Reports, ISQL, and many other programs. Client applications send and receive messages to and from ODBC, which in turn sends and receives them to and from the data requester.

Example

For example, if you are using the Manual Entry application to search for all assets where the manufacturer is FLUKE, you would first select **Search** to go to the **Find Asset** window, then enter FLUKE in the manufacturer field, and select **Search**. The Manual Entry application sends a message to ODBC, which in turn sends it on to the requester, to find all assets where the manufacturer is FLUKE. The requester then sends this message to the data engine over the network according to the network protocol. The data engine receives the message, searches the database file, and sends the answer back to the requester, again using the network protocol. The requester receives the answer, sends it to the Manual Entry application using ODBC, and Manual Entry displays the results on the screen for the operator.

Software Operation Summary

Every transaction between a Fluke Metrology application and the database involves the interaction of many programs. This holds true even if you are running both the client and server software on the same computer.

The machine running the data engine is not necessarily the file server on the network. For example, on the Fluke network there are UNIX computers that allow workstations to share files. When we run a Metrology application, one of the

workstations runs the database engine and becomes the database server. When this happens, the file server is not used at all.

As described in the above example, all communication for data travels between the requester and the data engine. Since the client software is running on a workstation, and the data engine is also running on a workstation, all communication occurs between workstations for database requests. The file server is still used, but only for shared files, such as reports, validation files and the software's executable files.

User Profile

Fluke Metrology Software is designed for five types of users:

1. System Administrator
2. Procedure Writer
3. Operator / Data Entry
4. Manager

At some sites, one person may perform all the functions, while at larger sites, separate individuals may perform the different functions.

The following paragraphs give general descriptions of tasks that the various users perform, which applications they have access to, and which chapters contain specific information about those tasks.

System Administrator

The *Administrator* is the computer or database expert for your site. There should be someone at your site who plays this role. This person understands how the lab's workflow operates, and, in fact, may have participated in its design and inception. The administrator has the big picture of the processes, and it is this person's task to customize the software and to define processes to assist the lab in meeting its goals.

The administrator is the focal point of the automation effort, because this person must integrate the software and the database into the existing system, and keep it running. From renaming fields through designing reports and backing up the data, the administrator must be aware of every aspect of the system. Each site needs to have one person who plays the various roles that the administrator is called on to play. The list shows the main responsibilities of the administrator.

- Add users, assign them their initial passwords, establish their access levels and designate segregation areas they are restricted to.
- Create and implement a backup plan for the database.
- Add instruments and maintain instrument configuration.

- Configure result templates and macros.
- Customize forms and reports to integrate the software smoothly into your organization's existing systems.

The administrator understands the philosophy behind the system, so this is the person who defines the data flow through the metrology process, customizes the data screens, performs data backup, and is responsible for the integrity of the data.

Normally, it is the administrator who installs the server and client software, adds and maintains users, migrates existing data, creates indexes, and works with ODBC and other local software.

None of these tasks are especially difficult. In all cases, Fluke has provided a default setup, philosophy, or process you can follow. Or, you can devise one of your own.

The Metrology Software system is designed with a sympathetic view of the magnitude of the administrator's job. Fluke provides, not just tools to aid in these tasks, but a built-in metrology system that will work for you immediately, with little or no customization required. In the beginning, you can concentrate on installation, data importation and training, and wait to do any customization until later, after you have gotten things under way.

The System Administrator has the highest user level.

Procedure Writer

The Procedure Writer creates and modifies procedures. The procedure writer has the following responsibilities:

- Writing MET/CAL or 5500/CAL Calibration Procedures
- Adding, deleting, and hiding procedures
- Hiding procedures
- Generating Test Uncertainty Ratio reports
- Testing procedures

The Procedure Writer can use the Manual Entry, Editor, and Run Time applications.

For specific information on the tasks of the Procedure Writer, see Chapters 2, 4, 6, and 8 of this manual (*Data Entry and Query Tasks, Running a Procedure, Using the Editor, and Managing the System.)*

Operator / Data Entry

Operators run procedures to perform verification and/or adjustment of a Unit Under Test (UUT). The operator has the following responsibilities:

- Calibrating UUTs using the Run Time application
- Saving, viewing and printing results

The operator can use the Run Time and Manual Entry applications.

For specific information, refer to Chapters 2, 4, and 6 of this manual.

Some sites have a workload heavy enough to support a separate individual just to input data. Typically, a data entry person would key in new asset or maintenance information. If all the assets in a system are labeled with barcodes, a data entry workstation would most likely include a bar code reader permitting some asset identification information to be wanded in, rather than entered using a keyboard.

In addition, if manual calibrations are performed, the technicians might fill out pertinent information on a hardcopy form that a data entry person would then transcribe into the database.

Manager

The *Manager* is mainly interested in looking at data. The manager uses the Manual Entry tool to review data and run reports.

To perform these tasks, the manager does not need to understand very much about the database or most of the applications.

Some managers may wish to create their own reports, and therefore will have to learn how to use a report writer. These managers will also have to understand data structures and how the data tables are related.

The User Interface

The Fluke Metrology Software is implemented as a group of Windows applications, using the same standards and conventions as other Windows applications. If you have some experience with Windows, you are already familiar in a general way with the user interface. Pull-down menus and fill-in dialog boxes are standard elements of this interface, and they operate in much the same manner from one application to the next.

Getting Data into the Database

Except for the automatically entered calibration results, when you provide information to the database, you do so by entering data into fields on the various forms.

A field is the smallest piece of data within the database. Fields make up records, and records make up tables.

The Database Tables

The software records asset information and calibration result in tables within the database. Usually, a form displays only one entry or record of a table. For example, each asset has one, and only one, entry in the Inventory table. When you view the information for that asset on the Inventory form, you are just looking at one of the many records contained in the Inventory table.

The main tables and their forms are listed below.

Inventory	Contains information about an asset, including asset number, model, manufacturer, and serial number.
Calibration	Each record contains information about a single calibration event, such as the calibration due date, the date it was actually performed, the temperature and humidity at the time of calibration, what standards were used, etc.
Calibration Results	Contains information about the results of a calibration. When a MET/CAL or 5500/CAL Calibration Procedure is running, FSCs that generate results automatically store the results in the database.
Location	Information about an asset, including its asset number, model, the manufacturer, and serial number
Repair	Each record contains information about a single repair performed on an asset, including the parts used in the repair, costs, failure description, and so on.
Customers	Contains information about customers such as name, address, shipping, tax and contract status.

Chapter 2
Backing Up Data

Contents **Page**

Vitally Important Information!

Backing up your data is of vital importance, and this chapter describes a minimum recommended plan.

Performing a backup of the database is not in any way complicated or difficult, and the tools provided make it relatively simple to do. However, backing up the data is of paramount importance. Here's why:

When you entrust your calibration data to a database, it represents one of your most important assets, because it is your only proof of the traceability of your calibrations. If your data should become lost or damaged, your business could be faced with a very difficult situation.

A database can be corrupted either by media failure or system failure. Both types of failure mean that the information is not accessible. Since the risk is unknown, the best practice is to be prepared.

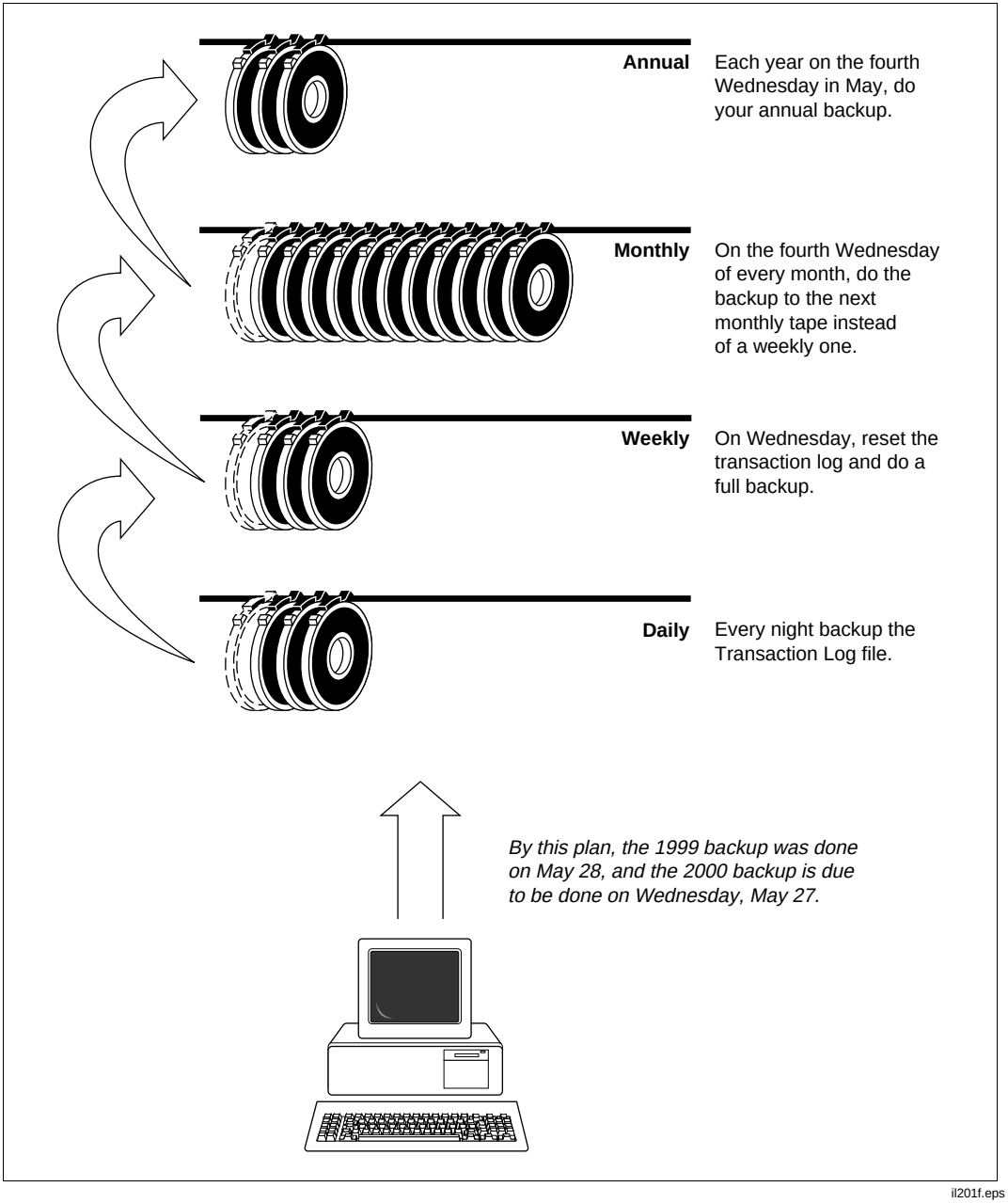
This chapter describes a minimum recommended backup plan for sites using Fluke Metrology applications to calibrate and manage their assets. This plan is based on using media (tapes, disks, CDs, etc.) that are systematically rotated so that, in the event of a catastrophe, you can recover your data with as little inconvenience as possible.

The importance of this activity is being emphasized because so much depends on your ability to recover gracefully in the event of an unforeseen occurrence. Once data has been lost, it can be very difficult to recover without an adequate backup.

Backup and Recovery Plan

You can back up data a tape drive, writeable CD-ROM or any media that you have easy access to by way of a locally available drive. The whole issue of backups and recovery boils down to the question of your own comfort level (or pain threshold). You must ask yourself what amount of inconvenience you are willing to withstand if you lose data. If you have no backup plan and your data is not available in hardcopy form, it could pose a serious problem for you, since you will have no way to recreate the data.

You should maintain separate daily backups during the current week, weekly backups for the current month, monthly backups for the current year, and annual backups that are all archived in a safe place. You can implement such a system by rotating disks or tapes. Assuming a five-day work week, you will need four tapes for daily backup (in this plan, there is no daily backup one day each week), four tapes for the weekly backup, 12 tapes for the monthly backup, and a tape for each year. Refer to the following figure as you read the description of the minimum recommended backup plan.



il201f.eps

Figure 2-1. Minimum Recommended Backup Plan

Daily Every day, make a backup of the log files. (The database automatically creates files describing its activity. See the Glossary Appendix for a more detailed description).

The daily transaction log file does not contain the entire database, but only a record of the transactions that have occurred since the last time a new log file was started.

Weekly On Wednesdays, do not do the daily backup, but a weekly. A weekly is a backup of the entire database, including the log files, the database files, and report files. This backup is made by over-writing the oldest weekly backup tape, so there is no Wednesday tape.

At the same time, restart the transaction log file. This is the mechanism that gives you, on a daily tape, the full set of transactions that have occurred since the last weekly backup was done.

Monthly On the last Wednesday of every month, you do a monthly backup. This is a backup of the entire database plus the METCAL directory on the server. To do this backup, exchange the newest weekly backup tape for the monthly backup tape, and over-write it.

Note that months sometimes have 5 Wednesdays, so you'll need at least 4 tapes for storing the weekly backups.

Annual On the last Wednesday of the year remove the newest monthly backup disk, tape or CD. Archive it permanently, and replace it with a new tape.

In every case, you are performing only a single backup operation. For example, you never need to do both a "daily" and a "weekly" backup, nor do you have to do anything special at the end of a month or year. Just rotate the tapes and make sure they are labeled correctly.

This plan is a practical one, and can be set up with very little bother. The entire rotation scheme can be done with as few as four tapes for the current week (Monday, Tuesday, Thursday, and Friday), three tapes for the current month (Week 1, Week 2, and Week 3), and 12 tapes for the current year. Rather than rotating the last weekly up to the month tier, there are advantages to maintaining the system with one tape for each month. One of these is that you do not need to remember which month has no tape.

This basic system can be modified to suit the workload at your site. For example, if you have a very heavy workload, you may choose to do full backups daily, in which case you would reset the transaction log file each day, too. In a very light workload situation, you might do full backups only once per month, backing up the transaction log file weekly.

Preliminary Checks

Before backing up your data you should verify its integrity. Computer systems can lose disk clusters. There are two steps to perform to verify the data.

First, verify the integrity of the disk. On Windows 95, Windows 98 and NT, run **scandisk**. If your data is stored on a network drive, you will not be able to run the utility. There may be a network utility that performs this function.

Second, run SQL Central or ISQL and use the Verify Database Integrity tool. After verifying the disk and the data, follow the backup plan. The actual steps you take in creating the backup are described following these preliminary checks.

Performing the Backup Using ISQL

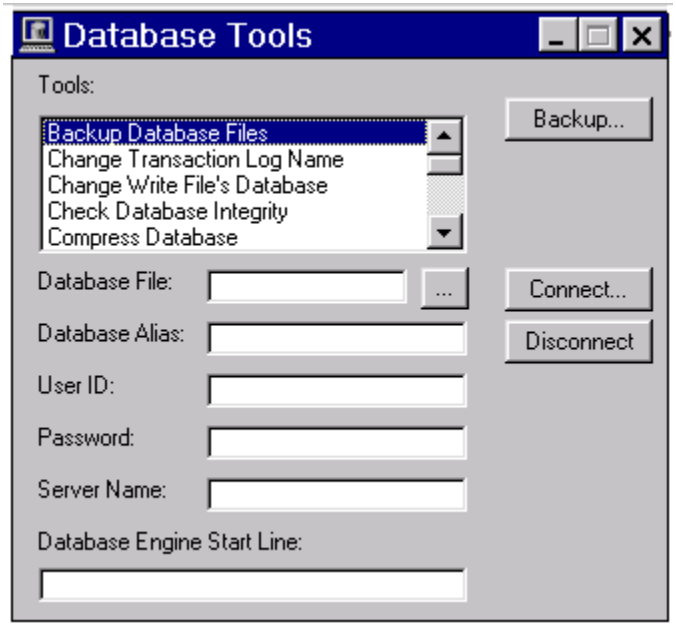
Before performing the backup, be sure to check that you are backing up good data. The ISQL application includes a complete set of Database Tools. One of them is dedicated to verifying the integrity of the database.

Verifying the Integrity of the Database

Make the following steps part of your routine backup activities:

1. Make sure no one is connected to the database.
2. Start the ISQL application by double-clicking its icon.
3. When ISQL starts, you can access the Database Tools in three different ways: double-click the Database Tools icon in the lower left corner of the window, select Database Tools from the Window menu, or use the keyboard accelerator Ctrl 4. Open the tools in any way you prefer.

   When the Database Tools open, you will see a window like the one shown in the following figure.



il204s.tif

Figure 2-2. ISQL Database Tools Dialog Box

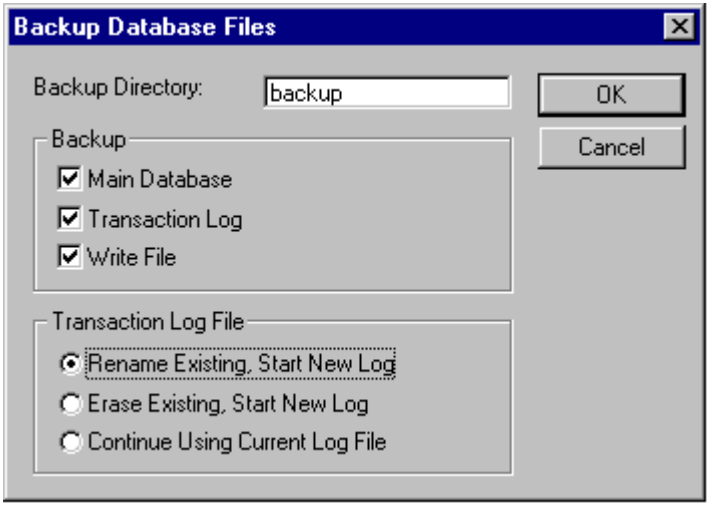
4. Scroll down the list of tools to Verify Database Integrity. Click on its name to select it.
5. Enter the database filename, and the MT User ID and password. The default password is “MT.”
6. Click the Check button. A dialog box opens asking if you wish to check only one table, or all of them. Click the All selection.

As the tool runs, it displays the names of the tables it is checking. Successful completion is indicated by “No errors reported” in the statistics window.

If a database file is corrupted, you will have to recover it from backup disks. Depending on the point in your backup cycle, you’d proceed to recover by first loading the most recent full backup, then applying today’s transaction log file.

Using ISQL to Create Backup Files

The ISQL Database Tools also provide assistance when it comes time to actually do the backup. Figure 2-3 shows the dialog box that appears if you select Backup Database Files. The selections shown on the figure are those you would make in preparation for a weekly backup.



zv203s.bmp

Figure 2-3. ISQL Database Tools Backup Dialog Box

The selections permit you to backup the main database, and/or the transaction log, and/or a write file if you are using one. Below that, you choose what to do with the Transaction Log File, whether to rename, erase, or continue using it. The backup utility in ISQL helps make data backup into a slight interruption, rather than an evening's effort.

Warning

Do not select Erase Existing, Start New Log if you have MET/REMOTE. Doing so will eliminate all data.

Copying the Backup Files to Archive Media

Regardless of the operating system, once you have created the backup files, the final step is to copy them to the archival media. In a very strictly controlled environment, you might even make two copies, one for storage off-site.

The nightly backup is called incremental because it contains only a record of the day's transactions, and it grows incrementally as the week goes on. If you follow the Wednesday evening plan described in this chapter, each Thursday morning you start with a fresh log file. On Thursday evening, the backup you do contains the day's transactions, then on Friday it has another day's transactions added. By keeping each of these, you can backup to whatever point might be necessary. One of the many handy utilities provided by SQL Central and ISQL lets you convert the transaction log file to SQL, so you can apply the changes to the most recent good full backup you have.

After you have done the backup, make it a practice to copy the backup files onto archive media as soon as possible. Archive these files:

incremental: transaction log

full backup: everything in METCAL and METDATA directories

Here are a few more points to keep in mind when you set up your backup plan:

- If you are using magnetic media (tapes or floppy disks), don't overwork the media by continuously overwriting it. Choose some number of overwrites as your limit, and maintain a tally either on the label or in your backup logs. When the media has been used the planned number of times, replace it.
- Be sure that your backups are stored safely. A good plan is to store all the backup tapes, disks or CDs off-site.
- It might be helpful to make duplicate copies of the monthly and annual backups. This way, you could feasibly restore the system without having to retrieve off-site media.
- There are many commercial backup software packages available. Make sure your backup software works! Several times a year you should restore from your backup system to verify its operation.
- Be sure to explore the Database Tools in SQL Central or ISQL. They provide features that can aid both the backup and any recovery you may have to do.

The database engine assists you in your preparations by making both backup and recovery as minimally stressful as possible. For example, it automatically recovers from many failures. If another type of failure occurs, you can recover quickly, *but only if you have been doing your backups!*

The Sybase SQL Anywhere on-line manual has full details about features built into the database engine to assist you in making an effective backup and recovery plan.

Chapter 3
Asset Management

Contents **Page**

Introduction

This chapter describes the asset management functions provided by Fluke Metrology software, such as adding a new asset or recording events manually.

When you perform an automated calibration, the Run Time application automatically records the event in the database. However, you may also have some manual calibrations to record, or you may want to manually record maintenance events.

Basic Database Operations

Manual data entry tasks are usually performed by operators, but the administrator should be familiar with these applications, too. Becoming expert in all of the applications and utilities can help administrators perform any needed customization more efficiently, and better prepare them to provide training and guidance to data entry and managerial users.

This chapter covers tasks related to data entry and querying the database. Note that any interaction with the database requires the database engine to be running.

The next few pages discuss basic operations that you must perform to access the database. All Fluke Metrology applications must connect to the database, so the first topic describes how to start an application.

Starting an Application

Each time you want to access the database for any reason, you must first connect to it as a data source. This is required by ODBC. The database engine itself controls access to the database

To connect to the database, proceed as follows:

1. Make certain that the Sybase SQL server (the Calibration Data Engine) is running. On networked systems, the Data Engine runs on the server.

   If you are on a single workstation system (not networked), you can simply observe if the Server icon is minimized, or check your Windows Task List.

   If the Data Engine is not running, start it. On Windows NT, Windows 98 and Windows 95, go to the Start menu, open Fluke Metrology Software, and select Calibration Data Engine.)

2. On a network Workstation, or on a Client, start the Calibration Data Requester, then start up the client application. It is not necessary to start a Requester on a standalone workstation.

3. When Sybase SQL locates the data source, it displays a dialog box asking for the user ID and password. Enter the name and password and press **ENTER**.

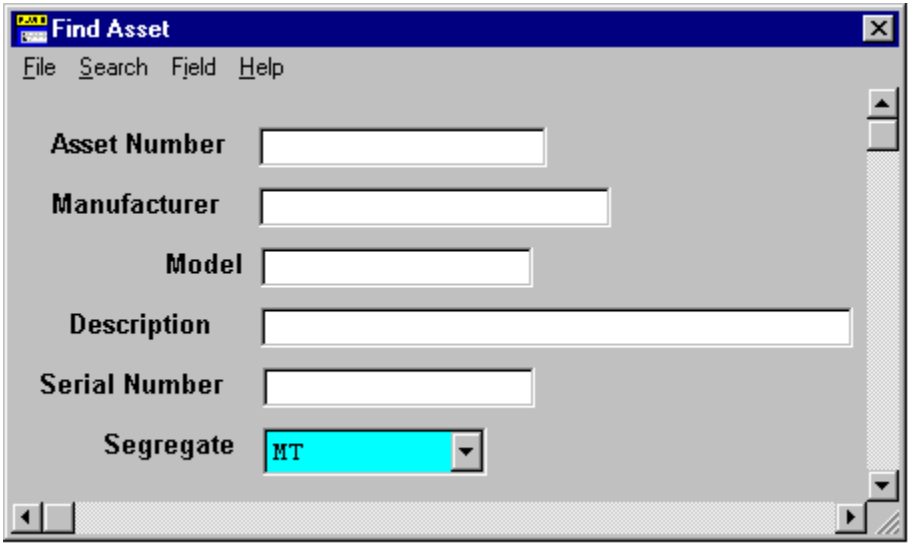
4. As shipped, the database has three valid user names, FLUKE, FLUKE3 and MT. The FLUKE user name has a password “FLUKE”, and the MT password is “MT.” You should change these passwords as soon as you are familiar enough with the software to perform those tasks.

When you are connected to the data source, the client application starts.

Browsing the Data

The Manual Entry allows you to view information in the database and reports.

1. Start the Calibration Database engine.
2. Start the Requester on a network client station (this step is not necessary on a standalone system).
3. Start the Manual Entry application
4. Select **Search**, then **Find Asset** from the Menu bar. This starts up a Search form, as shown in the following example:



zv306s.bmp

Figure 3-1. Search Form; Browsing the Data

5. In the Search form, enter query criteria in any fields. Criteria use the same ? wild card as the searching function within the Manual Entry application, and are logically ANDed in the same way.

   For example, to locate all the instruments made by Fluke, Flair, Flink, and Floorcrest having asset numbers beginning with 9, you could specify **FL?** as the manufacturer, and **9?** as the asset number.

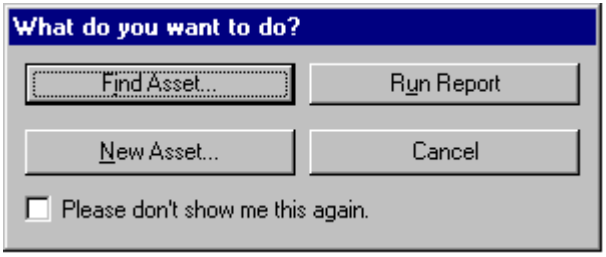
6. When you have filled in the criteria, click **Search.** The Manual Entry application pauses a few moments, changing the cursor to an hourglass. When the search is complete, you can once again access the Manual Entry application window.

You can browse data using the **Forms** menu the same way.

Adding a New Asset to the Database

To manually add an asset to the database, make sure that the Database Engine is running, then proceed as follows:

1. Start up the Manual Entry application and login to the database with a user name that has and access level of 3 or higher. (Note that you can configure access levels, so the access level you need to login under may be different, depending on your configuration.)
2. When the Manual Entry application appears, the following dialog box allows you to begin entering a new asset. Select **New Asset** from this dialog box.



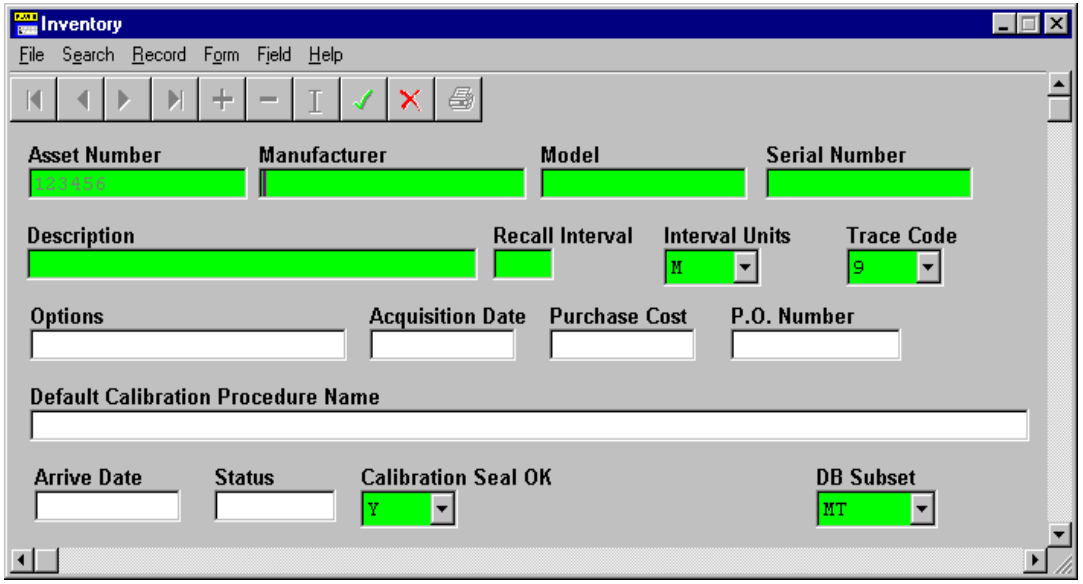
zv301s.bmp

Figure 3-2. Creating a New Asset

3. The next dialog box asks you to enter a new asset number.

   Type in a number that is not already in use as an asset number. (You may want to run a query or a report before starting to get a list of all the asset numbers you are currently using. These tasks are explained later in this chapter.)
4. If no message box appears warning that the number is already in use, an Inventory form displays similar to the one shown in Figure 3-3.

As shipped, the database requires entries in the fields that have green backgrounds on the Inventory form. Note that this list only gives the titles as they appear in an installation that has not been customized.



zv302s.bmp

Figure 3-3. Initial Inventory Form; Adding an Asset

The form you actually see may not be identical to this one. Remember that customization can make any of the forms appear completely different from those shown in this manual. You might title fields differently, move fields around on the form, and remove fields that are not required.

Recording Events

Each time an asset is calibrated, repaired, or relocated, information is gathered and entered into the database by way of the Calibration, Maintenance, and Location forms. For calibration and repair events, the Results form may also need to be updated.

Every event must be unique. This is ensured because the event's date and time are *core fields* in the database (see Glossary Appendix). The event's date, time, and asset number uniquely define a particular event.

Recording Maintenance and Location Events

To begin recording a repair or location event, use the Manual Entry application.

1. Under the Search menu, select **Find Asset** and indicate the asset number of the unit that has just been repaired or relocated. The application locates the asset and loads the Inventory form or the last form used.

2. Open the Maintenance or Location Form and enter the information called for.
3. Save each form. There is no "SAVE ALL" command to save all the forms at one time. Press the ✔button to save.

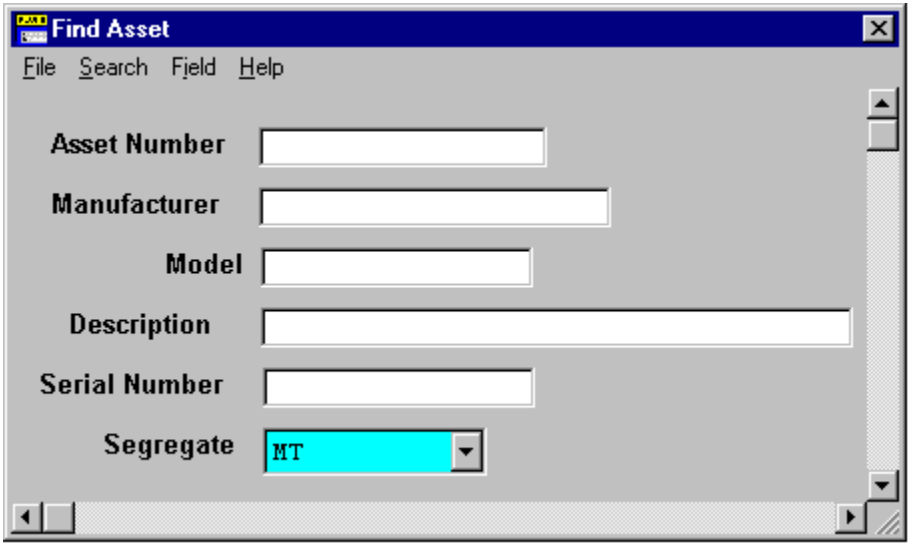
Manually Recording a Calibration Event

Each time an asset is calibrated, information can be gathered and entered into the database either automatically (as a direct consequence of running a MET/CAL procedure), or manually. The next chapter, "Running Procedures," discusses automatic recording of events.

Many times, however, you may wish to manually record a calibration event. You can do this while you are still setting up the database, as a way to have historical data brought into the database, or you may have other requirements for manually recording an event. For example, you may calibrate a piece of equipment using some other software package, or not using any automation at all. But, you still want to keep the data, and you want to be able to prove traceability. The Manual Entry application lets you enter information about calibration events into the database.

1. To record a calibration event, first locate the asset about which you wish to record data. Under the **Search** menu, select **Find Asset**.

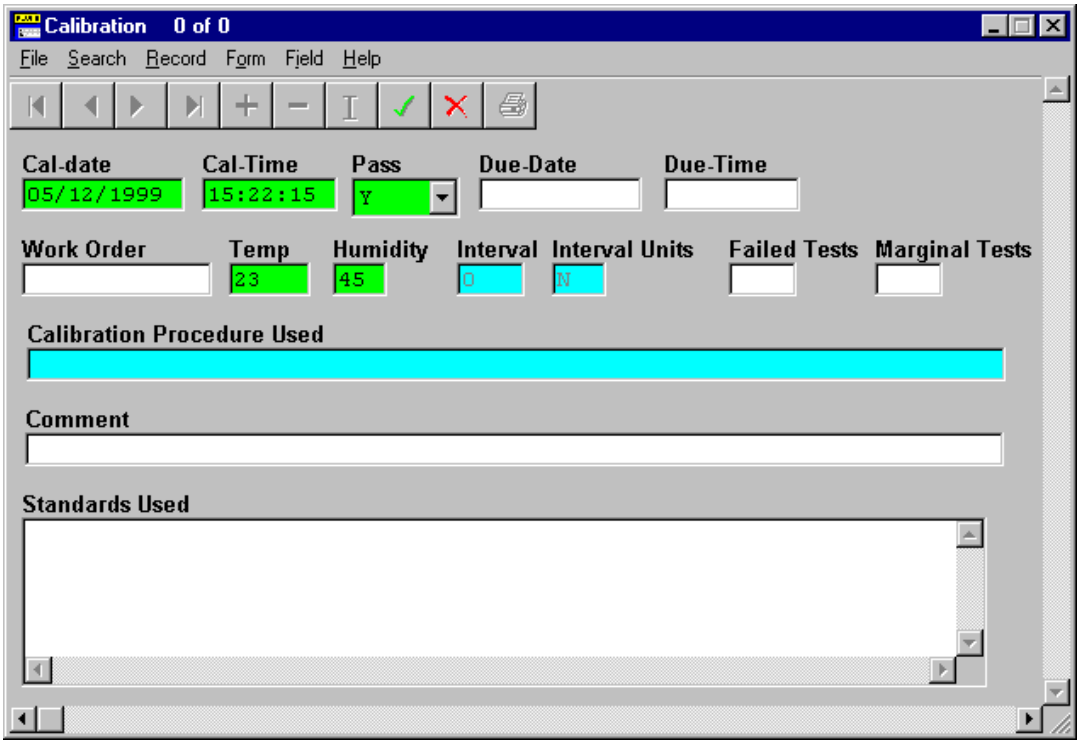
   In the following example display, asset number "SAMPLE-10" is selected. This is one of the sample assets provided for you in the database. As shipped, there are two calibration events in the database for SAMPLE-10. Type SAMPLE-10 under asset number. This will bring up an Inventory dialog box.



zv303s.tif

Figure 3-4. Looking up an Existing Asset

2. Next, open the **Forms** menu and choose the Calibration form. The following figure shows the resulting display.



zv304s.tif

Figure 3-5. Adding a Calibration Record

3. To enter **Standards Used**, go to the menu. Press the ✣ button or choose **Record** and **Add** to start a new Calibration form. Then, from the Calibration form select **Record**, then **Standards**.

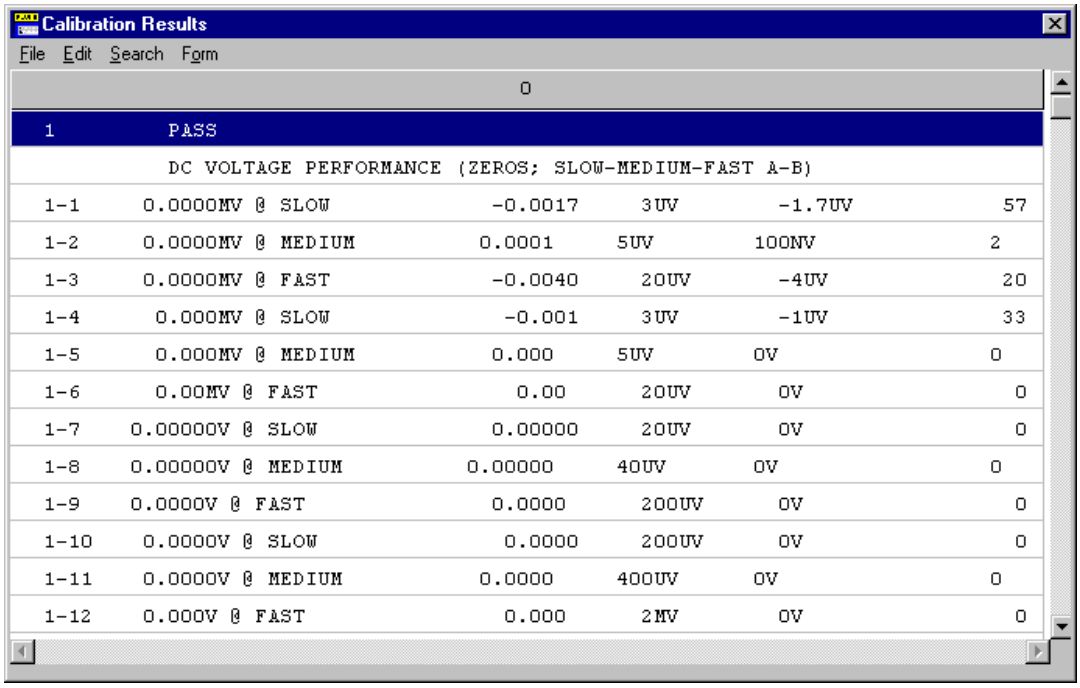
   A new dialog box appears. The top window shows the standards used by the current calibration. The bottom window shows standards that are available to be used. Assuming the outside vendor that calibrated the instrument was Fluke Corporation, select "FLUKE CORPORATION" from the bottom window by clicking the mouse to highlight it. Press **OK**. The standard will now appear in the upper window and on the calibration record.

   Note that until this has been accomplished, the standard is not part of the event. Also, the standard must already exist in inventory in order to be added to the Standards list.

Note

To represent outside vendor calibrations, an asset with the Asset Number of "FLUKE CORPORATION" is loaded in the Fluke Metrology Software database when it is shipped.

4. You can optionally record any test data and notes about the event in a separate Cal Results form. The following figure shows the Calibration Results forms.



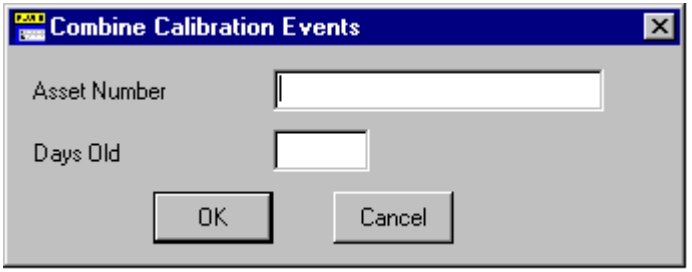
zv305s.bmp

Figure 3-6. Calibration and Results Forms

Combining Calibration Events

MET/CAL offers and advanced feature to combine calibration events. This feature is useful when you have a calibration event that is split into several parts.

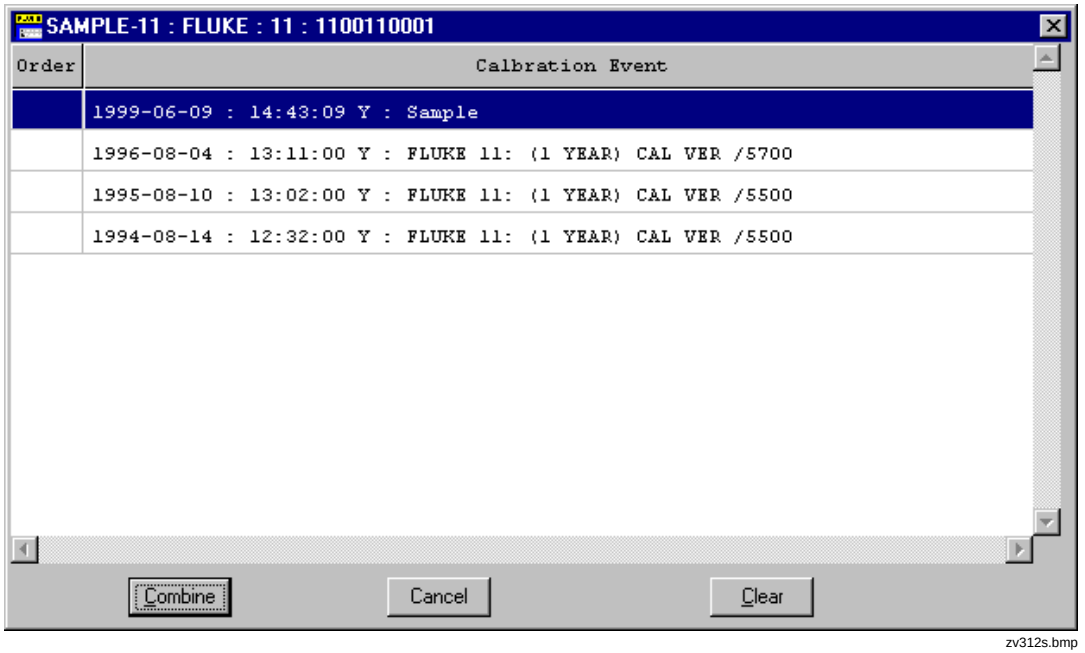
1. Make sure the Calibration Database is running and open the Manual Entry application.
2. From the menu, select **Tools**, then **Combine Calibrations**.
3. At the following screen, enter the asset number and how many days old the calibration events can be.



zv311s.bmp

Figure 3-7. Combine Calibration Events

4. After entering this information, select **OK**. This brings up a list of calibration events for the asset within the days old window.



zv312s.bmp

Figure 3-8. List of Calibration Events

5. Click in the order column to set the combination order for the calibration events.
6. Select **Combine** to combine calibration events.

There are special rules that apply to the combined calibration feature. They are:

- Use the combined cal feature when a new calibration event is created.

- The asset's most recent calibration even is used as the root summary for creating the new event, even if it was not selected in the order. The feature assumes it will be selected.
- If any one of the calibration events is a FAIL, the new combined record is a FAIL.
- In general, all of the values from the most recent calibration are used as the values in the new summary.
- The following fields are summed up from all the selected calibration events: 2350, 2351, 2352, 2353, 2324 and 2328.
- Standards are combined from all events.
- All standards must be in cal on the date of the most recent calibration event.
- The results are combined in the order specified.
- The operator has the option of deleting all the combined calibrations from the database.

Validation of Standards

The application validates a standard for the event only if that standard's calibration due date has not yet arrived. In the case of assets that require calibration according to number of uses, a calibration due date is not assigned until that number of uses has elapsed. Then, that date becomes its due date. In other words, it becomes immediately due for calibration on the date that it exceeds its number of uses setting.

If calibration time has been suppressed as a core field, it means that a given asset would not be able to be re-calibrated on the same day. Doing so would violate the rule that each calibration event must be uniquely identified, since the database uses Asset Number, Cal Date, and Cal Time as the unique identifier for each cal event.

This may be completely appropriate in your situation, or it may be a hindrance. Only the individual administrator can make that determination, because conditions and requirements vary. Topics like this one are discussed in Chapter 8, "Configuring the System."

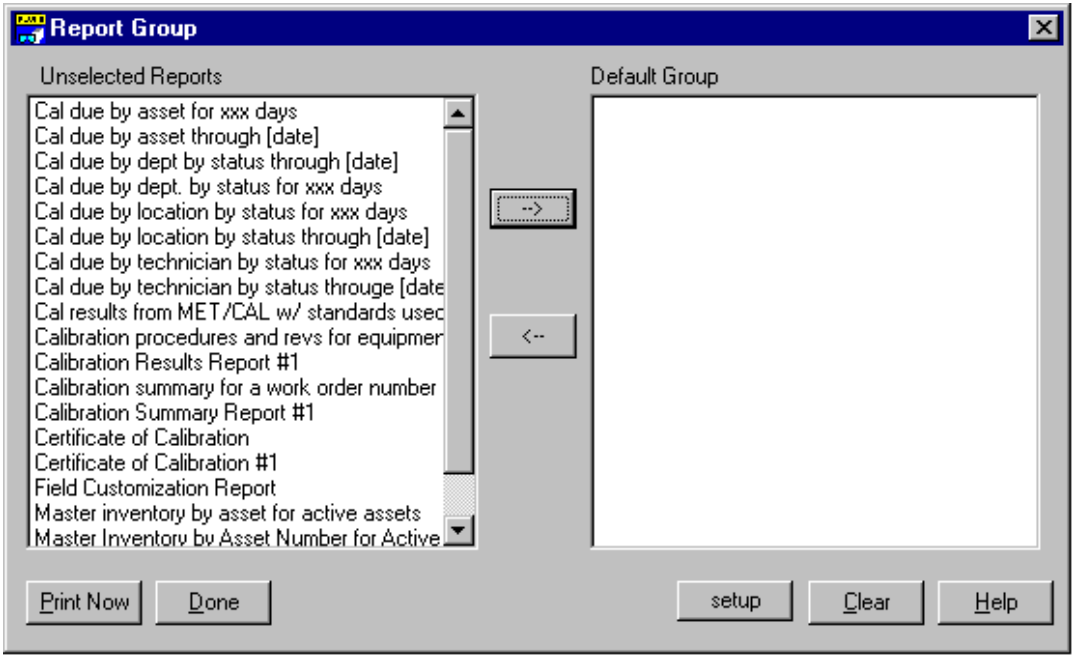
The Results form is tied to the Calibration event, so, to finish recording a calibration event, the operator may wish to open the Results form, as well as the Calibration form. The Results Form provides a kind of notepad for recording test results and other information about the calibration that you want to make part of the record for the event.

When a MET/CAL or 5500/CAL Procedure stores calibration results, it can generate a line of results data for each test. The file named RSLT_DB.FRM

formats these lines and stores them into the database field C2501, the Results field. For details, refer to Chapter 7, “Customizing the Run Time Application.”

Running A Report

1. Start the Manual Entry application, and log into the data source.
2. Select **Reports** from the Menu bar. Then select **Run Report**. A window lists the available reports on the left, and the currently selected group of reports on the right. See the following figure for an example of the Report Group display.



zv307s.tif

Figure 3-9. Example Report Group Display

Use the arrow buttons in the center to add reports to a group. If you want to run all the reports in a particular group, select the group. In the example shown in the preceding illustration, only one report is in the Default Group: a Master Inventory report that lists all the active assets in the database.

3. Select **Print Now** to preview and print the report(s).

The Manual Entry application only lists reports that have already been added.

Chapter 5 contains detailed information about reports, including tutorials on how to create and modify them.

Deleting a Calibration Record

CAUTION

Deleting a calibration irretrievably removes information from the database.

Only level 4 and 5 operators can delete records from the database.

To delete a calibration event, view it on the Calibration Form and select **Delete**.

The record of that calibration is ***irretrievably removed from the database.***

Purging an Asset

CAUTION

Purging an asset irretrievably removes information from the database.

Only level 4 and 5 operators can purge records from the database., It is not possible to purge an asset that is used as a standard.

To purge an asset, view it on the Inventory form and click **Purge**.

The asset being viewed and ***all of its history are irretrievably deleted***.

If the asset you want to remove from the system has been used as a standard for the calibration of another instrument, then the application will not permit purging the asset.

Automated Data Entry

You can enter information into the database by manual entry, or you can speed things up by automating the data entry. This section describes how to prepare information for automated entry, and the steps to follow to bring the information into your database.

For adding small amounts of information to your database, such as a single calibration or repair event, or a couple of assets, the most appropriate handling is by way of manual entry using the forms. But, for large amounts of data, or for routine, scheduled transfer of information to or from some other program, the automated data entry option is probably better. It takes less time, and avoids typographical errors.

Using the automated method of getting information into the database is identical to typing the information into the forms; it just happens very quickly.

The source of data for importation is an ASCII text file. Importing from a file is faster and more reliable than manual entry. Importation behavior is identical to manual entry. That is, all the rules for manual entry and the customization of fields apply.

Two situations call for importing data:

- Database downloading, a one-time importation of your existing database after you first install the Fluke Metrology software.
- Routine data entry, such as importing calibration history records after they have been collected in a file.

A one-time importation from another computer system requires some knowledge of the database structures. Give some serious thought to how your database system will integrate with the Fluke database before you start downloading data.

Importing an Existing Database

If you have an existing database of metrology information, you may only be interested in doing a one-time importation of your historical data. The next few paragraphs describe the major steps to prepare the data and bring it into the Fluke Metrology database.

Step 1: Get The Data To The PC In ASCII Text Form

To import data from another computer, you must first place the data in a format that the PC can read. The data must be ASCII text, written to MS-DOS compatible floppy disks or to the hard disk.

Your current system may be able to save your data to a file in ASCII text format. ASCII text is just alphanumeric text with no special control or hidden characters. Most systems can output ASCII text, but if your current system cannot, or if you are not sure how to do it, consult with your local experts (usually your MIS or Data Processing group or perhaps a PC/Microcomputer Support group). Fluke Product Support may be able to help.

Once you have created the ASCII files, record them to a DOS-formatted disk.

Step 2: Decide if You Will Import History

You need to decide how much of your old data to migrate into the new database. Importing your entire database is more involved if you want to include all history records.

If possible, it is recommended that you make a clean break with your old system. You can do this by importing only inventory data and the most recent calibration history record for each asset. That greatly simplifies the importation process. Additionally, your old data may have erroneous entries that will not be accepted, or that may present problems later. For example, traceability may not be guaranteed for older calibration records.

If you choose to include data older than the most recent record in the importation, more than one text file is required. Import inventory data first, separate from calibration or other history data. Create a separate text file for calibration and

history data. The files containing old historical records must also include the asset number.

The actual format of the text file is covered later in this section.

Files Used In Importation

The following types of files are associated with data importation:

Source The Source file contains your data. This is the text file that has already been discussed. The fields can appear in any sequence.

Definition The Definition file specifies the format of the data in the Source file. This is a file that you must create. More information on this topic follows in this section.

Error The Error Log file, IMP_LOG.TXT, contains a list of any errors (and the offending records) that occur during importation. The importation program generates the Error Log in the directory with the executable program. The importation program also produces a second Error Log file that contains the rejected lines for reimportation after you have corrected the errors. The first of these will be named IMP_REDO1.TXT. Rejected lines from subsequent importations will be named IMP_REDO2.TXT, IMP_REDO3.TXT, etc., unless you delete these files as soon as you correct and reimport them.

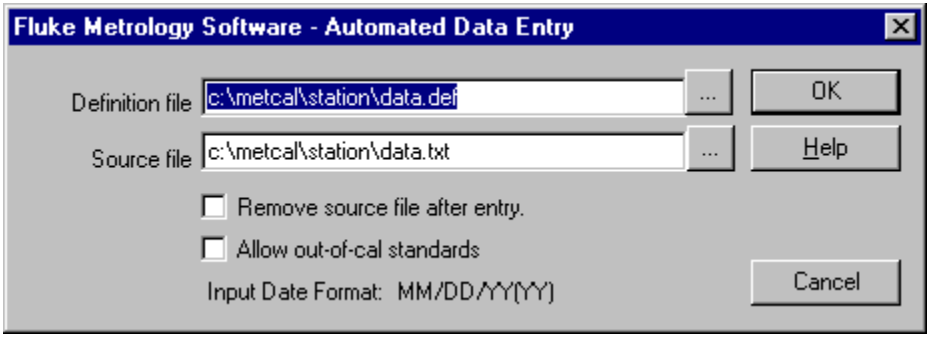
Initialization The Initialization file, METCAL.INI, located in the Windows directory, contains entries that control the default Definition and Source file names and the default field separators. These can be modified if necessary.

Speed of Importation

Importation of large files can take a long time, especially if you have many validation files. Validation is active during importation. While it may occasionally seem to be an inefficient process, automated data importation is much faster and more accurate than manual entry.

Data Importation Procedure

1. To start the import process, first make sure the SQL Database Engine is running, then select the Import icon from the Fluke Metrology Software programs and double click.
2. After you login, the Import screen will appear. Select **IMPORT** and the prompt screen will appear, as shown in the following figure.



il309s.bmp

Figure 3-10. Automated Data Entry Prompt Screen

3. Enter the filename of the definition file and the source file into the prompt screen.

   a. If you plan to delete the source file after import, select the "Remove source file after entry" check box. This is useful for keeping historical data current if you are importing data on a recurring basis.

   b. The "Allow out-of-cal standards" checkbox is used during initial importation of data. This selection turns off the automatic standards checking feature. This feature would cause errors if calibration histories of standards should be imported after the history of a UUT using that standard. This selection can be used only by level 5 operators.

4. After entering the filenames and (if desired) enabling one or both of the options, click **OK** to begin the import.

5. If errors occur during import, an error screen will appear. If this happens, open the error log to view the errors.

Specifying the Definition File

When you type in the name of the Definition file, include the path if the file is not located in the current (default) directory.

You can specify a default entry for the Definition file in the import section of the METCAL.INI file (located in the Windows directory).

```
import_def = <filename>
```

Example

```
import_def=C:\METCAL\DEFILE27.DEF
```

The line above in the METCAL.INI file specifies "DEFILE27.DEF" as the filename for the import Definition file. When the operator performs an import,

that filename is displayed in the prompt. If a different Definition file is required, the operator can simply type over the displayed value.

No error checking is performed on this value at start-up time. Thus, if the file does not exist when the Automatic Data Entry application starts, the operator will not get an error message until the Definition file is used. Examples of valid filenames are “import.def”, “c:\a\b\c\d\abc.xyz”, or “\cal\metcal.def.” The default could also be just a path to which you would add the filename at import time.

Specifying the Source File

In the box for the Source Data file, be sure to include the path if the file is not located in the current directory.

Again, you can specify a default Source filename entry in the METCAL.INI file:

```
import_src = <filename>
```

EXAMPLE:

```
import_src=C:\METCAL\SOURCE27.SRC
```

This line in the METCAL.INI file specifies “source27.src” as the filename for the import Source file. When the operator performs an import, the prompt box displays that filename. If a different Source file is required, the operator can simply type over the displayed value. No error checking is performed on this value at start-up time. Thus, if the file does not exist, the operator will not get an error message until the Source file is used.

The Error Log File

The Automatic Data Entry application creates an Error Log file assigned the default name IMP_LOG.TXT.

If the application encounters any errors during import, it displays screen messages and may copy bad records to the Error Log file.

More information about these files and the importation process follows.

The METCAL.INI Initialization File

The file METCAL.INI contains a section used by the Automatic Data Entry program. This portion of the file can be identified by the heading “IMPORT.”

```
[IMPORT]
import_src=C:\METCAL\SOURCE27.SRC
import_def=C:\METDATA\DEFILE27.DEF
delimiter = “
separator = ,
```

The last two lines indicate the default field delimiter and field separator. These can be modified if these characters are used within the data to be imported, such as in a note or comment field. More information on the use of these fields follows.

The Definition File

The Definition file contains information to tell the importation program how to read the Source file(s). The Definition file is an ASCII file that you create. In its simplest form, it is just a list of the fields to be imported and keywords.

Normally, you will create a particular Definition file only once. Once you have it, you use it once to download your data, or you use a different one repeatedly to import data generated by ongoing operations. The latter typically deals with data in a specific format, which in turn allows the continued reuse of a single Definition file.

A Definition file contains the following:

1. One or more keyword(s).
2. Field number lines in the order the fields appear in the Source file.

Creating a Definition File

To create a definition file, you must first understand a few simple rules about the data to be imported.

1. Use an ASCII text editor such as Notepad to create or modify the definition file.
2. The beginning of the file must contain the keyword(s) identifying the type of data to be imported.
3. The first field must be the asset number field, 4201.
4. The field numbers must match the data fields to be imported. That is, if the second field in the data source file is Manufacturer, the matching definition file entry must be 4202.

The remaining lines consist of matching field number, comments or the SKIP key word.

Key Words Used in Definition Files

Key words are special words used in the Definition file to control the flow of the import.

Data Type Keywords

You must enter key words at the beginning of the Definition file. Only one key word is allowed per line.

INVENTORY	Specifies that the source file contains data to be imported into the Inventory table.
CALIBRATION	Specifies that the source file contains data to be imported into the Calibration table.
REPAIR	Specifies that the source file contains data to be imported into the Repair table.
LOCATION	Specifies that the source file contains data to be imported into the Location table.
CUSTOMER	Specifies that the source file contains data to be imported into the Customer table.

Special Key Words

SKIP	Specifies that the matching field in the data file contains data that is not to be imported. This keyword can be used in combination with any of the above keywords and multiple times in a definition file.
COMMENT LINES	Any line beginning with a semicolon (;) or pound sign (#) is a comment line and is ignored. Do not use the ";" or "#" in place of the SKIP key word.

Example Definition Files

The following sample Source file contains several records, one per line. As you can see, the fields are separated by commas (the default character).

```
645878,Function Generator,2123,,Philips,PM5134
758400,Oscilloscope,5634,403,Philips,PM3350
589538,DMM,7384,02A,Fluke,8506A
993743,DMM,6587,059,Fluke,8842A
```

Notice that the preceding data belongs in Inventory, and that the nature of this data means that this example importation creates new assets. A Definition file to import the above Source file would appear as follows:

```
INVENTORY
4201
4204
4218
4207
4202
4203
```

The keyword “INVENTORY” is associated with all of the fields of the Inventory screen. Other keywords are necessary for importation to the Calibration, Maintenance, and Location screens.

The Source File

The Source file is the ASCII text file that contains the data you wish to import to the database. Source files are in a free (not fixed) format.

Source files are character-delimited fields. That is, the data is simply listed with commas (or some other character) separating each field. This is the easiest format to use, and is one that most systems can output. The Definition file (the file that the Importation application uses to understand the format) can then be as simple as a list of the fields.

In the following example, each asset is described in one record and each record is on its own line. Note the varying record length (line length) in this example of a free format file delimited by commas:

```
645878,Function Generator,2123,,Philips,PM5134
758400,Oscilloscope,5634,403,Philips,PM3350
589538,DMM,7384,02A,Fluke,8506A
993743,DMM,6587,059,Fluke,8842A
```

The Source file must also meet the following requirements:

1. There can be only one record per line.
2. A record or line may be a maximum of 2,000 characters in length.
3. The Asset Number is a required field regardless of the type of data to be imported, even though Asset Number is only found in Inventory. For example, a Source file containing Location history data would also need the asset number to identify the assets in order for them to be properly recorded into the database. The Asset Number field must be the first field in a record.

The SKIP Keyword

The file shown in the previous example consists only of a list of the field numbers. Ordinarily, nothing more is needed. An import could be performed with those files.

However, you can use the keyword SKIP to bypass the data item. You can use this keyword for initial import of historical calibration records from an existing database, where the standards’ existence, record sequence, and due-dates may prevent normal importation.

Creating the Data Source File

The data source file is an ASCII file with comma-separated fields. Quotation marks are used to delimit data within a field. For example, if a note or comment field contains a comma or quotation mark, the entire field can be framed with quotation marks to preserve the contents of that field.

Example:

```
1234,10/10/1995,"In spec, .001" OK"
```

A matching definition file might be:

```
CALIBRATION
4201
2301
2326
```

The line would be interpreted thus:

```
ASSET NUMBER        1234
CALIBRATION DATE    10/10/1995
CAL  NOTE           In spec, .001" OK
```

Data Source Conversion

The data to be imported must meet the format, type and field length specifications of the Fluke Metrology database. In this example the source data file contains the words "PASS" and "FAIL to indicate a calibration status. The other applications expect this data to be a "Y" or "N" in field 2323 to indicate a pass or fail condition. The source data input might look like this:

```
1234,10/10/1995,PASS
```

The third field must be converted to a "Y" in order to be imported into field 2323. The following Basic program can perform translation of an ASCII file to prepare it for importation to the Fluke database.

```
CLS
REM The next two lines prompt for input and output file names
INPUT "Enter filename for filtering"; filename$
INPUT "Enter filename for output"; filename2$
REM the next two lines open the input and output files
OPEN filename$ FOR INPUT AS #1
OPEN filename2$ FOR OUTPUT AS #2
REM Beginning of a "do" loop, the other end is the "loop while"
DO
REM Take data in from the input file
INPUT #1, data1$            REM Gets Asset number
INPUT #1, data2$            REM Gets Cal Date
INPUT #1, data3$            REM Gets Pass/Fail
```

```
IF data3$ = “PASS” THEN data4$ = “Y”  REM Translate Pass
IF data3$ = “FAIL” THEN data4$ = “N”  REM Translate Fail
REM Print data back to a new file and replace comma field separators
PRINT #2, data1$; ",";data2$;”,”;data4$
LOOP WHILE NOT EOF(1)
REM Close the files before ending
CLOSE #1
CLOSE #2
```

This program would produce the following output:

```
1234,10/10/1995,Y
```

A BASIC program could also add a required field that is missing from a source file. In this case the Trace code, field 4209, is missing. The field can be added

```
CLS
REM The next two lines prompt for input and output file names
INPUT "Enter filename for filtering"; filename$
INPUT "Enter filename for output"; filename2$
REM The next two lines open the input and output files
OPEN filename$ FOR INPUT AS #1
OPEN filename2$ FOR OUTPUT AS #2
REM Beginning of a "do" loop, the other end is the "loop while"
DO
LINE INPUT #1, data$
REM   Take in a line from the input file
REM   Print the line back to output file. DO NOT add line feed
REM   or CR (this is done by adding ";" at the end of the line)
PRINT #2, data$; ",";
REM   Add a "5" to the line, could be a "TRACE" code.
PRINT #2, "5"; ",";
REM   Add an "M" to the line, could be an interval type.
REM   No ";" at the end of this command terminates the line
PRINT #2, "M"
LOOP WHILE NOT EOF(1)
REM   Close the files before ending
CLOSE #1
CLOSE #2
```

Testing the Importation

You may use testing when you initially import the database. Testing is useful when creating new Definition files for use with standard Source files generated by your calibration system.

Because importation can take a long time if there are very many records, it is a good idea to first run a test. Choose a reasonable number of records, small enough for quick importation, but large enough to catch structural and systematic errors.

This way you can identify and correct errors that would otherwise cause the Error Log file to become large and waste valuable time.

After you have run a test import, examine the Error Log file with a text editor. Check it for any errors that may have occurred, and make any corrections to the Definition or Source files as necessary.

Also, be sure to look at the test data using the Manual Entry application after testing the importation. If you specified the wrong field sequence or column numbers, data placement or truncation errors might have occurred. These are logic errors that would not necessarily be caught by import error checking.

After you make the necessary corrections, perform the full importation.

Chapter 4
Running Calibration Procedures

Contents **Page**

Safety Considerations

Calibration systems may generate or measure dangerous voltages. Both the operator and procedure editor must be aware of possible hazards.

To avoid electrical shock and fire hazards, follow the instructions listed below.

⚠Warning

Never use a screen saver with MET/CAL or 5500/CAL Run Time Applications. The Fluke software has been designed to promote safety by displaying a high-voltage safety symbol when dangerous voltage is present.

A screen saver program hides this symbol, so should never be used on workstations where calibrations are performed.

⚠Warning

Avoid electric shock by always following these rules:

- **Use interconnect wiring and adapters rated for the highest voltage and current levels carried.**
- **Never touch exposed conductive portions of signal interconnect wiring (even after instruments have been turned off.)**
- **Only touch the insulated portion of a connector. First disconnect the interconnect wiring at the source, then at the measurement side.**
- **Always use insulated adjustment tools.**
- **If the calibration program terminates abnormally, verify that front panels of the source instrument indicate standby mode.**

Introduction

This chapter describes how to use the Run Time application.

You can skip this chapter if you have only MET/TRACK licenses.

The Run Time application is the central tool provided by the MET/CAL and 5500/CAL licenses, because this application executes procedures to actually calibrate instruments.

Each time a calibration begins, the procedure can present prompts for the operator to provide needed information, such as the asset number of the instrument being calibrated, or the temperature or other environmental factors. During the calibration, the screen might display illustrations showing test points, or forms providing an opportunity to enter a value, such as a UUT reading.

At the conclusion of a Calibration Procedure, you may see other prompts or displays, and you may be able to direct the system regarding how to deal with the output of the tests. You may be able to generate a Calibration Certificate or other report, or you may even be given the opportunity to discard the results if the test failed. All these possibilities depend on the particular way your system administrator has configured and customized the Fluke Metrology applications for your site. If you are not the administrator, be sure to speak with that individual before beginning any calibrations, so you know what to expect and how to respond appropriately.

This chapter can be helpful whether this is your first experience with the Run Time, or whether you have previously worked with Fluke Calibration software.

You need a basic knowledge of the Windows graphical interface; for example, how to operate pull-down menus and dialog boxes, how to make selections from list boxes, and so on. This chapter does not provide any instruction in these topics.

This chapter assumes that your system is completely configured, and that you have installed some Calibration Procedures.

It is helpful if at least one person at your site has gone through all the exercises in the *Getting Started Guide*. If you are not the same individual who did the Getting Started exercises, consult with that individual or with your system administrator to find out any configuration details you may need, and to learn which Calibration Procedures and reports are available to you.

The remainder of this chapter discusses the following major topics:

- Overview of the Run Time Application
- Selecting a Calibration Procedure
- Executing a Calibration Procedure
- Post Run Results

Overview of the Run Time Application

The Run Time application allows you to select a procedure and run a calibration on a Unit Under Test (UUT). Figure 4-1 gives a high-level overview of how the Run Time application works when you run a procedure.

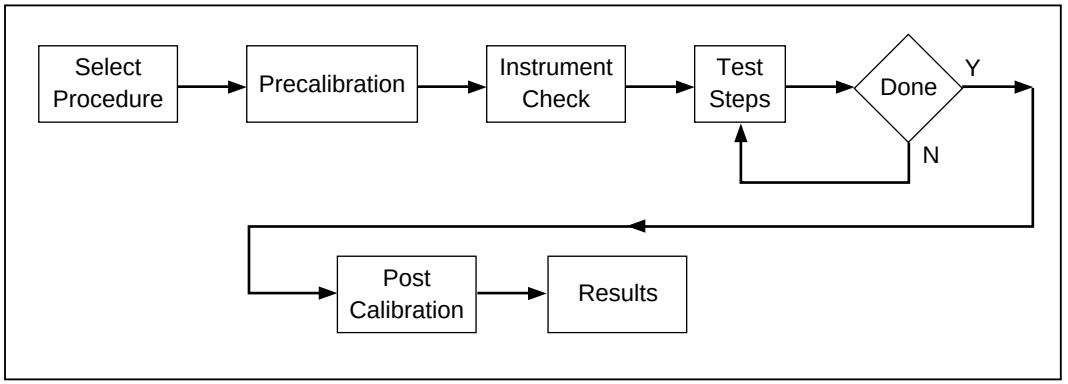


Figure 4-1. High-Level Overview of Run Time

The purpose of the Run Time application is to execute Calibration Procedures and save calibration results. Some of the information in this chapter requires an understanding of the Fluke procedure language. Therefore, this chapter includes some information about the MET/CAL procedure language.

Note that, while it is not necessary to understand the procedure language in order to use the Run Time application, an understanding of the basics of the language helps you understand how the Run Time application functions and, in particular, how its operation may vary depending on the procedure being executed. Refer to the *MET/CAL Reference Manual* for additional information about the MET/CAL procedure language.

Sequence of Events

This section presents a sequential outline of the major steps performed by the Run Time. Most of these steps are described in more detail later in the chapter.

To execute the sequence of steps outlined below, start the Run Time and choose **Calibrate** in the top-level menu. It may be helpful to actually run a procedure while reading this information to obtain a full understanding of the process.

Note

The first step in the process, selecting a procedure, is initiated by the operator. All the remaining steps are done by the Run Time application, although some of the steps require operator involvement.

Step 1 - Select Procedure

There are two ways to select the procedure: by name, by asset. The procedure selection method is configurable (by going to the Run Time menu and selecting **Configure**, then **Procedure Selection Method**).

For more information refer to the heading later in this chapter titled, “Run Time Menu.”

Step 2 - Check User Permissions for Required Database Fields

The Run Time verifies that the user has permission to write to all required fields in the Calibration and Inventory database tables.

Step 3 - Load Procedure

The selected procedure and all required subprocedures, if any, are loaded into memory. No operator interaction is required in this step, but the Run Time displays a sequence of one or more pop-ups that trace the loading of the procedure.

Step 4 - Load Pictures

If there are any picture (.bmp) files required, the Run Time verifies that the specified files exist.

Step 5 - Check Instrument Configuration

Next, the Run Time verifies that the system instruments required by the selected procedure are configured.

If a required instrument isn't configured, it can be added to the configuration by choosing **Instruments** in the **Configure** menu. Verifying that required instruments are configured does not involve any communication with the actual instruments.

Step 6 - Check Traceability

For each required system instrument, user-configured system instrument, or manual standard, a traceability check is done to make sure the instrument is in cal.

ALL the instruments used in a calibration are considered to be “standards”, and therefore all of them must meet the following rules:

1. The standard must be in calibration before the instrument that is the current UUT. For each standard, only the most recent record meeting this criteria is used.
2. Every standard must have passed its most recent calibration. (Field C2323 = Y.)
3. The standard’s due date + due time must be in the future, beyond the due date and time of the current UUT. If the standard’s due date + due time is blank, then due date + due time is assumed to be infinity; it is never due for calibration.

Step 7 - Prompt for UUT Asset Number

Now the Run Time prompts for the asset number of the UUT.

For more information refer to the heading later in this chapter titled, “Asset Number Prompt.”

Step 8 - Check User Permissions for Database Asset Information

The Run Time verifies that the currently logged-in user has permission to write to those database fields that the Run Time automatically updates for the specified asset. This verification is needed because MET/CAL automatically writes to the database at the conclusion of the calibration.

Step 9 - Pre-Prompt Dialog

A dialog prompts the user for associated information before procedure execution starts. This dialog is often used to prompt for temperature, humidity, and work order number. The dialog is configurable, however, and can be customized for each site.

If you do not want to use the pre-prompt dialog, it can be turned off.

For more information, refer to the heading later in this chapter titled, “Prompt Dialog.”

Step 10 - Create Test Results Window

The Test Results window is a separate window that shows a one-line, color-coded result line for each test step in the Calibration Procedure.

For more information refer to the heading later in this chapter titled, “Test Results Window.”

Step 11 - Reset Internal Data

This step prepares the system to execute the selected procedure. No interaction with the user is involved.

Step 12 - Initialize IEEE-488 Boards

This step is skipped for 5500/CAL, since it does not support IEEE-488 interfaces.

Step 13 - Open Required Serial Ports

The Run Time next opens all serial ports referenced by the selected procedure.

Step 14 - Test Required System Instruments

The Run Time performs an instrument-specific test for each required system instrument.

If a system instrument fails its test, it is probably because it isn’t turned on or is not set to the configured address. In some cases, however, a system instrument may be broken, or the PC’s IEEE-488 bus or serial ports may be locked up or otherwise malfunctioning. If the instrument is turned on and set to the correct address, you may wish to reboot the PC before concluding that the instrument is malfunctioning.

For system instruments configured on serial ports, the port configuration (specified using the Windows Control Panel) must agree with the instrument settings, and the cable must be of the correct type and pinout.

No test is done for the UUT, for user-configured instruments, or for manual standards.

Note

A user-configured instrument is an instrument used as a calibration standard, and remotely controlled by MET/CAL, but for which there is no built-in driver.

Step 15 - Clear Required System Instruments

This step returns all required system instruments to a known, reset state prior to procedure execution.

Step 16 - Execute Procedure

After performing the foregoing preliminary steps, the Run Time now executes the selected Calibration Procedure.

A Calibration Procedure is a sequence of calibration test steps. Each test step involves setup steps followed by an evaluation. In many cases the setup steps involve prompts to the operator, and the evaluation may also involve an operator prompt.

The sequence of steps executed and the types of prompts displayed to the operator are controlled by the MET/CAL procedure.

To execute a procedure, MET/CAL starts at the first procedure statement and executes statements in the order they occur, except when one of the following two things occurs:

- the procedure jumps to a different step, or
- the operator uses the Post Test dialog to alter the execution sequence.

The operator's ability to alter the execution sequence of the procedure can be disabled by the procedure writer.

Each test step generates a test result. By default, the results of each test step are shown to the operator in the Post Test dialog. However, it is common for procedures to be written so that the Post Test dialog appears only after a test step fails.

More information on procedure execution, including descriptions of the different types of tests and operator prompts, is provided later in this chapter. Refer to the heading titled, "Procedure Execution."

Step 17 - Clear Required System Instruments

This step returns all required system instruments to a known, reset state following procedure execution.

Step 18 - Close Required Serial Ports

All serial ports referenced by the selected procedure are closed.

Step 19 - Automatic Prompt to Operator: Remove All Connections

An operator prompt display reminds the operator to remove all connections. You cannot configure or inhibit this prompt. It is a built-in feature of the Run Time application.

Step 20 - Post-Prompt Dialog

A dialog display prompts the user for associated information after procedure execution ends. This dialog is often used for displaying status information (such as the number of marginal PASS results, elapsed time, and so on), as well as to prompt for a remark that will be stored with the calibration results.

You can configure the post-prompt dialog, and customize it for each site. It can be configured independently, based on whether the overall result of the calibration is PASS or FAIL.

If you do not want to use the post-prompt dialog, it can be turned off.

For more information, refer to the heading later in this chapter titled, “Prompt Dialogs.”

Step 21 - Post Run Results

The Post Run Results dialog allows the operator to view or print the test results.

Test results are normally saved automatically, and no operator intervention is required. The dialog includes three buttons which offer the following choices:

Save Calibration Data

Discard Calibration Data

Discard All Data for UUT

The following aspects of the Post Run dialog can be configured using the MET/CAL Run Time application:

- The dialog can be turned off.
- A macro can be set up to view and/or print specified reports.

  Up to 8 different calibration reports can be set up for viewing or printing. By default, the following reports are provided:

  1. Certificate of Calibration
  2. Calibration Results
  3. Calibration Summary
- Users can add new reports or modify existing reports as needed.
- Users can selectively disable the software’s ability to save test results, discard test results and discard all information for the UUT.

After the operator chooses **Close** to exit from the Post Run Results dialog, the Run Time returns to its top-level menu and waits for further input from the user.

At this point, the Run Time has completed the sequence for running one procedure, and the user can execute a different procedure, re-execute the same procedure, or exit from the application.

Run Time Menu

This section provides information about the Run Time application's menu items. When investigating the menu, it may be useful to open the Run Time and actually view the various items available.

File

Choosing **File** in the top-level menu causes a drop-down submenu to display. The menu has only one item, **Exit**. Choosing **Exit** closes the MET/CAL Run Time application.

There are three other, equivalent ways to close the application:

1. Choose **Close** in the system menu. The system menu drop-down can be seen by clicking in the upper-left hand corner of the window.
2. Type **Alt** + **F4**.
3. Click on the **X** in the upper, right-hand corner of the window.

The Run Time application cannot be closed during procedure execution.

Calibrate

Choosing calibrate initiates the sequence of steps described under the heading above titled, "Run Time Overview."

The first thing the user sees after choosing **Calibrate** is a dialog that allows the procedure to be selected.

Actually, one of two different dialogs will be seen, depending on the procedure selection method. The Options menu item allows you to select procedures. A description of the Options menu appears a few paragraphs forward.

Selection By Procedure Name

If the procedure selection method is *By Procedure Name*, choosing **Calibrate** causes the Run Time to display the Procedure Selection dialog. This dialog presents a directory of available MET/CAL procedures, listed by name.

You can add, delete, or modify procedures using the Editor. Open the **Tools** menu and select **Administrator**. The Run Time application does not provide a way to add, delete or modify procedures. It only allows the operator to select a procedure to be executed.

The Procedure Selection dialog includes OK, Cancel and Search buttons.

- **OK** selects the highlighted procedure. If no procedure is highlighted, an error message displays, and you can try again.
- **Cancel** returns to the top-level menu without doing anything.

- **Search** allows searching for any substring contained in a procedure name.

The Procedure Selection dialog does not list hidden or non-executable procedures. Hidden procedures are usually subprocedures that are not selected directly. Use the Editor to work on hidden or non-executable procedures.

Selection By Asset Number

If the procedure selection method is *By Asset Number*, choosing **Calibrate** causes a prompt for the UUT asset number to appear. The number you type into the dialog box must be a valid asset number.

After the user enters the asset number of the UUT, the Run Time queries the database to find the procedure associated with the specified asset.

Options

Choosing **Options** causes a drop-down submenu to display. The submenu contains the following items:

Demo Mode

The Run Time can operate in two modes: *demo* mode and *real* mode.

If the *Demo Mode* menu item is checked, the Run Time is in demo mode; otherwise, the application is in real mode.

In demo mode, the Run Time application does not communicate with instruments. Also, in demo mode, the application doesn't open or close serial ports, doesn't initialize the IEEE-488 bus, and doesn't read from or write to serial or IEEE-488 instruments.

In all other respects demo mode is the same as real mode, including the saving of results. In general, results generated when running in demo mode should be discarded, since demo mode is provided for testing and debugging only, and is not intended for use in calibrating real UUTs. Note that you can debug procedures by using the Editor's Test Run capability. The Run Time's demo mode may be useful, however, in testing prompt screens, Run Time reports, and so on.

When the Run Time is in demo mode, it appends " `- Demo Mode` " to the application name in the window title.

It is not possible to change modes during procedure execution.

Procedure Selection Method

Highlighting *Procedure Selection Method* in the Options menu causes an associated menu containing the following choices to be displayed:

- By Procedure Name
- By Asset Number

At any given time, one and only one of these items will be checked. You can click on a different item to change the selection method. The Run Time remembers the selected method until you change it. Choosing a different procedure selection method has no effect until you choose **Calibrate**, in the top-level Run Time menu.

Refer to the heading above titled, "Calibrate" for a description of how the procedure selection method affects the Run Time operation.

Instruments

This selection lists the current configured instruments in your system and allows you to add new instruments as the need arises. It also provides a means for editing or deleting instruments.

IEEE-488 Boards

This selection allows you to configure the number of IEEE-488 boards that are in the computer and lists their manufacturer model name.

Default Serial Port

This selection allows you to configure a default serial port for serial controlled UUTs.

Default Font

This selection allows you to determine the font used in the Editor application, and the Run Time application's Post Test - List option. To maintain column alignment when procedure listings are viewed, the default font must be a fixed-point font (one in which every character consumes the same amount of space on the line).

Display Font

You may configure the font in some of the prompts and messages Run Time operators see during procedure execution. The Display font may or may not be a fixed point font. Verify that the fonts you select display correctly with the monitor used on the calibration system.

Post Run Reports

Configuration of Post Run Reports permits choosing those reports in MET/CAL's RPT directory that you wish to make available to the Post Run or Autosave Macros which run at the conclusion of a calibration.

Post Run Dialog

The Configure Post Run Dialog selection allows the restriction of an operator's options after the procedure is complete. For example, if you want an operator to have only the option of viewing a report, select View and click OK. After the UUT is tested, an operator can only view the report, not print it. If none of the options are enabled, the Post Run dialog box does not display at the end of each procedure.

The Post Run dialog can be set up so the results of a calibration are NOT stored automatically, and even to initially select "Discard Data on Close." These may be useful features during development or testing of new calibration procedures, or when you are first getting MET/CAL set up.

Autosave Pass / Autosave Fail Macros

The Autosave pass and Autosave Fail macros let you determine which reports are displayed or printed automatically when the procedure completes. The list is the same one as for the Post Run Macro, created from those reports you select using Configure Post Run Reports.

Post Run Macro

Configuring the Post Run Macro lets you set up a macro for the operator to use at the end of each calibration procedure. The operator executes the post Run Macro by choosing the MACRO button on the Run Time Post Run dialog box. For example, if you want the operator to be able to both view and print the reports, select both **Print** and **View** for each of the listed reports. The list is the same one as for the Autosave Pass and Autosave Fail macros, created from the reports you select using Configure Post Run Reports.

Break

Break has only one purpose: it aborts the operation of a closed-loop procedure. A closed-loop procedure runs without operator intervention. *Break* is normally disabled (grayed out) except during procedure execution.

Tools

You can configure the Tools menu to contain any other applications you want to start from the Run Time. By default, the menu includes other MET/CAL applications.

When you select an application in the Tools menu, the Run Time attempts to start that application. If the selection is another MET/CAL application that requires logging into the database, you do not need to re-enter the login information (database, name, and password). The other application starts using the same database, login name and password as the user who is currently logged in to the Run Time.

There is no automated way to modify the contents of the Tools menu using the MET/CAL applications. To add, change, or delete items in the Tools menu you must edit the MET/CAL initialization file, `metcal.ini`. The Run Time Tools menu items are in the [Run Time Tools] section. Use Notepad or a similar text editor to modify the initialization file.

Security

Choosing **Security** in the top-level Run Time menu causes a drop-down submenu to appear. The submenu has only one item, *Password*.

Password

Choosing **Password** causes the Change Password dialog to be displayed.

This dialog allows the currently logged-in user to change his or her password.

About

Choosing **About** causes a dialog to display that shows:

a) Company Name

b) Copyright Message

c) Application Name

d) Product Name and Version

e) Current User
This is the full name (not the login name) of the currently logged in user.

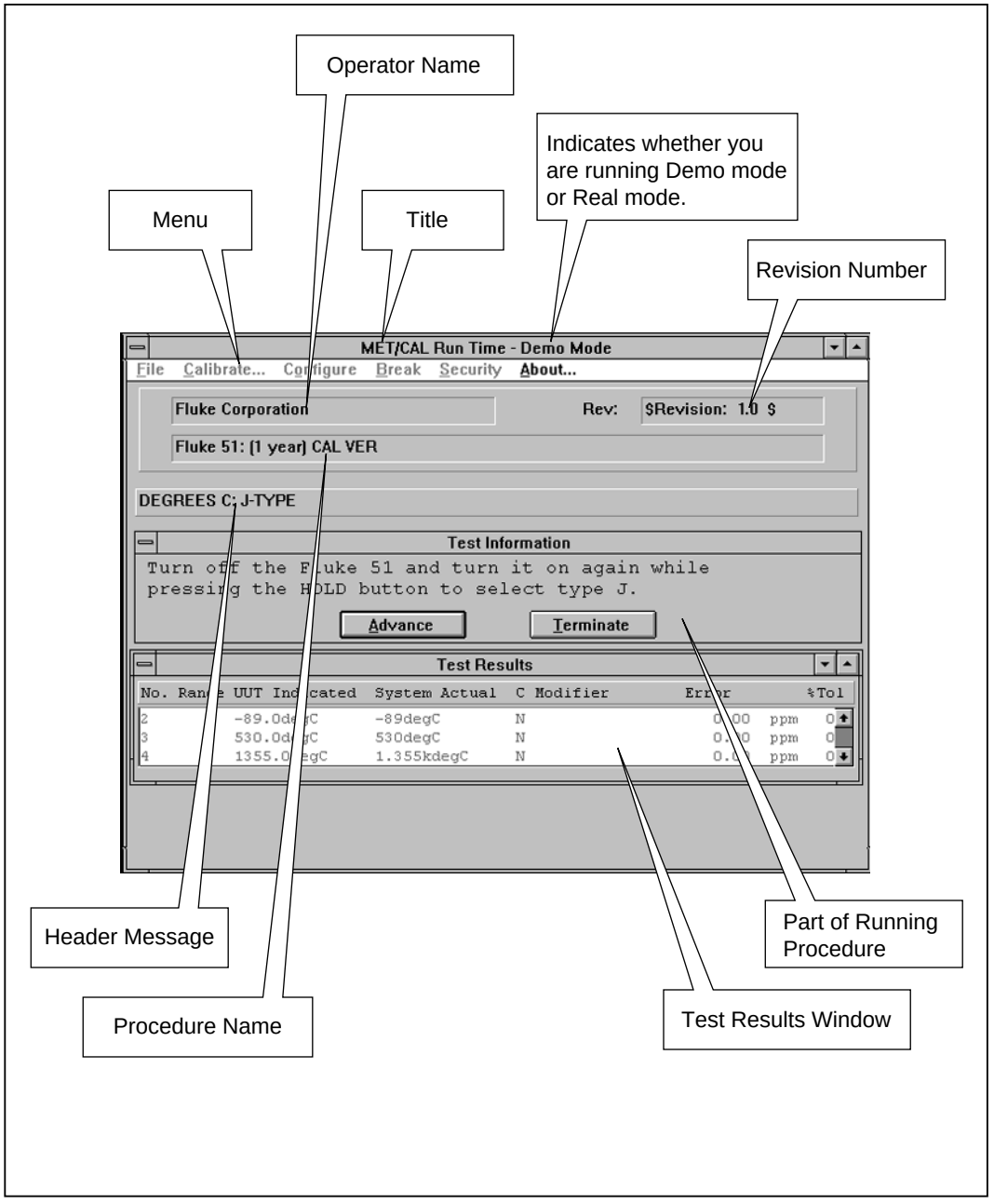
The Run Time Window

The top-level window of the Run Time application consists of a title, a menu (described above), and information displayed in the window below the menu. The information displayed in the window usually changes as the procedure runs.

The information described in this section is not the only information presented to the operator during procedure execution. Other information, however, is not written directly on the Run Time's main window. Rather, it appears in various pop-up dialogs and windows that appear and disappear as the procedure runs. Most, but not all, of these pop-ups require a response from the operator.

Refer to Figure 4-2 as you review the following information about the Run Time display.

il402c.eps



il402c.eps

Figure 4-2. Run Time Window

Title

The window title shows the name of the application (MET/CAL Run Time). If the Run Time is in demo mode, "`- Demo Mode`" is appended to the application name in the title.

Menu

See preceding discussion, under the heading, "Run Time Menu" above.

Run Time Window Information

In the upper portion of the window, below the menu, the Run Time shows the operator's name, the procedure revision, and the procedure name.

Operator Name

The name shown is the full name of the currently logged in user.

Procedure Name and Revision Number

The procedure name and revision are not shown unless a procedure is running.

If the main procedure calls a subprocedure, the name and revision of the subprocedure are shown while the subprocedure is executing. After the subprocedure returns to the main procedure, the main procedure's name and revision are displayed again.

Header Message

Below the procedure name, space is reserved for a header message. This message is generated whenever a HEAD statement in a procedure is executed. A HEAD message remains on the screen until a subsequent HEAD statement changes it, or the procedure terminates.

An empty HEAD statement in the procedure can be used to clear the message.

The header message can be used to provide a descriptive header that groups the individual test steps of a procedure into functional groups. For example, a procedure to calibrate a handheld DMM might divide its test steps into AC voltage tests, DC voltage tests, AC current tests, DC current tests, and resistance tests. The procedure could include a corresponding HEAD statement for each group of tests. The operator will then see, for example, "AC Voltage Tests" displayed in the header message area during the ACV tests, "DC Voltage Tests" during the DCV tests, and so on.

System Stimulus/Measurement Information

Below the area reserved for the header message, a line is reserved for information about the current output stimulus or current measurement. This area is not seen on the illustration, since it is hidden at the moment by the Test Information dialog.

For example, if you write and execute a procedure containing the lines:

```
5700 1V S
DISP Pause
```

you will see the text: **Stimulus: 1V** appear in the Run Time screen when the 1V stimulus is present. The stimulus is present from the time the setup statement executes until the 5700A is reset.

If you write and execute a procedure containing the line:

```
3458 1V S
```

you will see text similar to **Measurement: 1.001V**. This text is shown until the operator responds to the prompt that instructs him to adjust the UUT for a reading of 1V. The *1.001V* here is just an example. If you actually do this and run the procedure in real mode you would see the actual HP 3458 measurement here.

Asset Number Prompt

Each MET/CAL Calibration Procedure verifies or adjusts a particular Unit Under Test (UUT). The asset number of the UUT is required by the Run Time because calibration results generated during procedure execution are stored in a database in which all information is organized by asset number.

If the asset number does not exist in the database, a separate dialog prompts for all required fields in the Inventory Table. The Run Time application then writes the new asset and all required inventory table data to the database.

If the procedure selection method is *By Asset*, then, when the operator enters an asset number, MET/CAL retrieves the name of the associated procedure from the database.

If the specified asset is new to the database, no information exists for the asset in the database, so the associated procedure name cannot be retrieved. In this case, MET/CAL displays the procedure selection dialog to allow the procedure to be selected by name. If the procedure name field in the inventory table is configured as a required field, the specified procedure name is pre-loaded into the subsequent prompt for required inventory data. This mechanism allows the association between asset number and procedure name to be initially set up using the Run Time application when the asset is first added to the database. The other way to do this is to use the manual Entry application to add the asset, including the associated procedure name, before using the Run Time application to run a procedure to calibrate the asset.

If the specified asset is not new to the database, inventory table information for the asset already exists in the database. In this case, as stated above, MET/CAL attempts to retrieve the name of the associated procedure from the database. If there is no associated procedure name, MET/CAL displays the procedure

selection dialog to allow you to select the procedure by name. The Run Time application cannot, however, modify an existing inventory table entry, so the selected procedure is executed by the Run Time, but the name is not saved in the inventory table for the specified asset. The only way to permanently associate a specified procedure name with an existing asset is to use the Manual Entry application.

Prompt Dialogs

There are two configurable dialogs that normally are presented to the operator each time a procedure is executed. These dialogs are referred to as the pre-prompt dialog and the post-prompt dialog. The pre-prompt dialog displays after you select a procedure, but before procedure execution begins. The post-prompt dialog displays after procedure execution terminates.

Pre-Prompt Dialog

You can configure the pre-prompt dialog that appears after a procedure has been selected, but before the start of procedure execution.

Used the pre-prompt dialog to display read-only information, or to prompt the operator for information that will be written to the results or, in some cases, used during procedure execution.

The pre-prompt dialog is controlled by the parameter named *prompt* in the [startup] section of the MET/CAL initialization file, metcal.ini. The *prompt* parameter specifies the name of the file that controls the pre-prompt dialog.

The default initialization file contains the line:

```
prompt = promptdb.txt
```

This indicates that the name of the file that controls the pre-prompt dialog is promptdb.txt.

To turn off the pre-prompt dialog, the system administrator can change the initialization file to read, “prompt = none.”

An equivalent way to turn off the pre-prompt dialog is to edit the prompt file (default name `promptdb.txt`) and change it so that it contains no lines other than comments and blank lines. Any line that begins with a pound sign (`#`) or a semicolon (`;`) is a comment line.

The default pre-prompt dialog prompts the operator for the temperature, the humidity, the work order number, and asks whether the calibration seal on the UUT is intact. By modifying the pre-prompt file the administrator of the system can prompt for any desired information.

If the value entered by the operator for a data item is to be written to the database, the value must be written to one of the fields of the calibration table, and it must not be a field that is automatically updated by the MET/CAL Run Time.

Normally all fields in the pre-prompt dialog are displayed in a single dialog. However, if the system is set up to prompt for a large number of items, a sequence of separate dialogs will appear until all necessary prompts have been shown.

Refer to the heading "Customizing Prompts for the Operator" in Chapter 7 for additional information, including a detailed description of the format of the pre-prompt file, and other information about customizing the pre-prompt file.

Post-Prompt Dialogs

There are two post-prompt dialogs: the pass-prompt dialog and the fail-prompt dialog. After each procedure execution one or the other, but not both, of these dialogs is displayed.

If the overall result of the procedure is PASS, the pass-prompt dialog is displayed. If the overall result is FAIL, the fail-prompt dialog is displayed. The condition that determines an overall result of FAIL is that one or more of the individual tests has failed because the UUT is out of tolerance at that test point.

The pass-prompt and fail-prompt dialogs are separately configurable.

Use the post-prompt dialogs to display read-only information, or to prompt the operator for information that will be written to the results.

The pass-prompt dialog is controlled by the parameter named *pass_prompt* in the [startup] section of the MET/CAL initialization file, `metcal.ini`. The *pass_prompt* parameter specifies the name of the file that controls the pass-prompt dialog.

Similarly, the fail-prompt dialog is controlled by the parameter named *fail_prompt* in the [startup] section of the MET/CAL initialization file, `metcal.ini`. The *fail_prompt* parameter specifies the name of the file that controls the fail-prompt dialog.

The default initialization file contains the following two lines:

```
pass_prompt = pass_db.txt
fail_prompt = fail_db.txt
```

These indicate that the names of the files that control the pass-prompt and fail-prompt dialogs are `pass_db.txt` and `fail_db.txt`, respectively.

To turn off the pass- or fail-prompt dialogs, you can change the initialization file to read, "pass_prompt = none", or "fail_prompt = none." Since the two dialogs are independently configurable, it's possible (for example) to set up the system so that a post-prompt dialog appears when the overall result is FAIL, but not when the overall result is PASS.

As with the pre-prompt dialog, an equivalent way to turn off the post-prompt dialog is to edit the appropriate prompt file (default names are `pass_db.txt` and `fail_db.txt`), and change it so that it contains no lines other than comments and blank lines. Any line that begins with a pound sign (`#`) or a semicolon (`;`) is a comment line.

The default pass-prompt dialog shows the operator the number of marginal tests, the time it took to calibrate the UUT, the overall test result (PASS, FAIL, etc.), and asks the operator to indicate that the results are either “As Found” or “As Left”, and to enter an optional comment.

Note that a “marginal test” is one where the test passed, but the error expressed as a percentage of the test tolerance is greater than the adjustment threshold. The adjustment threshold is a configurable parameter set in the procedure. It is typically set to 70%.

The default fail-prompt dialog is the same as the pass-prompt dialog, except that it also shows the total number of tests that failed.

However, as with the pre-prompt dialog, the post-prompt dialogs can be configured by the person who sets up the system. When you run a procedure, you may or may not see the information and prompts described for the default prompt dialogs.

If the value entered by the operator for a data item is to be written to the database, the value must be written to one of the fields of the calibration table, and it must not be a field that is automatically updated by the MET/CAL Run Time.

Normally all fields in the post-prompt dialog are displayed in a single dialog. However, if the system is set up to prompt for a large number of items, a sequence of separate dialogs will appear until all the prompts have been shown.

Refer to the section “Customizing Prompts for the Operator” in Chapter 7 for additional information. There, you will find a detailed description of the format of pass-prompt and fail-prompt files, and other information about customizing them.

Procedure Execution

This section describes how the Run Time application executes a MET/CAL procedure. The information following is mainly intended for the person who will be writing those procedures, or modifying a supplied procedure.

Some of the information in this section is repeated in the *MET/CAL Reference Manual*. In fact, as you prepare procedures for your site, refer to that manual for complete reference information about the FSCs that make up the MET/CAL procedure language. The descriptions that follow are provided to assist in giving an overview of the Run Time's procedure execution process.

Procedure Statements

Each MET/CAL procedure consists of a header followed by a sequence of procedure statements.

An individual procedure statement begins with a step number, followed by a Function Selection Code (FSC). The FSC is a keyword that identifies the type of procedure statement.

Example:

```
1.001  5500   40   35.00V      0.72U      5kH      SI      2W
```

In this statement the FSC is **5500**, which identifies the statement as one that controls a Fluke 5500A calibrator.

Example:

```
1.001  DISP          Set the Capacitance Decade to 4.70nF.
```

In this statement the FSC is **DISP**, which identifies the statement as one that issues a prompt to the operator.

For a detailed description of the procedure header, see the *MET/CAL Reference Manual* under the heading, "MET/CAL Procedure Header."

Step Numbers

As mentioned above, each procedure statement begins with a step number. The step number has two parts, as follows:

- The major step number is the test number.
- The minor step number defines the sequential position of the statement within that test.

You can see step numbers in a procedure by using the Editor application, or looking at the currently executing procedure using the Post Test List option.

Example:

```
3.020  5500       3.513V                50H       SI    S  2W
```

The above example is the 20th statement in test number 3.

Tests

Statements in a procedure are grouped by major step number into blocks that comprise individual tests.

A typical test has three parts:

1. Connections
2. Setups
3. Evaluation

In a simple case, a single procedure statement performs all three parts of the test. For example:

```
STEP  FSC  RANGE NOMINAL   TOLERANCE   MOD1   MOD2  3  4 CON
1.001 5700 400   350.0mV   2.9U        60H                2W
```

This single statement (assuming the default state of the ASK flags), performs the following operations:

1. **Connection** - An automatic connection message displays that prompts the operator to connect the UUT to the 5700A Output Terminals.
2. **Setup** - An automatic range message displays that prompts the operator to set the UUT to the 400mV AC Range.
3. **Setup** - The 5700A is programmed to output 350mV, 60Hz with internal sense.
4. **Setup** - The *Slew* dialog displays that prompts the operator to *Adjust stimulus for a UUT reading of 350mV at 60Hz*. When slewing is complete, System Actual is set to the final 5700A output voltage, and UUT Indicated is set to the value in the Nominal field.
5. **Evaluation** - UUT Indicated is compared to System Actual, according to the specification in the Tolerance field, and test results are generated.

In many cases, however, a test is made up of multiple procedure statements. More information about the three parts of a test (connections, setups, and evaluation) is given below.

Connections

The operator performs connections under the direction of prompts that the procedure generates. Connection message prompts can be automatically generated by an instrument FSC or the connection (CON) FSC, or the person who writes the procedure can create the message prompts using the display (DISP) or picture (PIC) FSC.

Connection messages instruct the operator, or, in the case of the PIC FSC, show the operator graphically how to interconnect the UUT and the calibration standards. See *Automatic Connections*, *Operator Information Prompt*, and *Pictorial Information Prompt* in the section on run time prompts later in this chapter.

For a complete descriptions of the **DISP**, **PIC**, and **CON** FSCs see the FSC Reference chapter of the *MET/CAL Reference Manual*.

Setup

Once the connections are made, a test typically requires one or more setups to be performed. Setups fall into one of three categories:

1. Setting up the UUT.
2. Setting up the standards required for the test.
3. Setting up the memory registers containing UUT Indicated and System Actual, which are used to perform the test evaluation.

Setting up the UUT

The UUT is set up either manually by the operator or automatically by MET/CAL if it has remote control capability. Prompts to the operator to set up the UUT can be automatically generated by an instrument FSC or RNG FSC or by using the DISP, MESS, PIC, or SET FSCs. For example, the following statements would all generate the operator prompt *Set UUT to the 10 MOhm range.*

```
1.001  RNG          10MZ
1.001  5700   10    10.00000MZ     0.00410U              2W
1.001  DISP         Set UUT to the 10 MOhm range.
```

See the headings, *Automatic Range Message*, *Automatic Test Setup Prompts*, *Automatic UUT Setup Prompts*, and *Operator Information Prompt* in the section on run time prompts later on in this chapter. For a complete description of instrument FSCs, **RNG**, **DISP**, **MESS**, **PIC**, and **SET** see the FSC Reference chapter of the *MET/CAL Reference Manual*.

The IEEE FSC is used to set up UUTs with an IEEE-488 interface and the PORT FSC is used to set up UUTs with an RS-232 interface. The statement below uses the SCPI remote protocol to program a UUT with an IEEE-488 interface.

The UUT is a DMM. The SCPI commands select 2-wire Ohms and the 10 MOhm range. The IEEE-488.2 ***OPC?** (Operation Complete query) command is also sent to determine when the setup is complete.

When the setup is complete, the DMM returns a "**1**" which MET/CAL subsequently reads using the **[I]** special construction. The programming string is automatically terminated by MET/CAL using the IEEE-488 EOI (End or Identify) line.

```
1.025  IEEE          FUNC "RES";:RES:RANG 10e+6;*OPC?[I]
```

If the UUT has a RS-232 interface, the same command could be sent using the PORT FSC. For serial UUTs the terminator must be specified. The terminator can be included at the end of the output string or declared globally using the [OTERM] special construction (see Port FSC in FSC references). In this case a carriage return is specified at the end of the output string using the **[13]** special construction.

```
1.025  PORT          FUNC "RES";:RES:RANG 10e+6;*OPC?[13][I]
```

For a complete description of **IEEE** and **PORT** see the FSC Reference chapter of the MET/CAL Reference Manual.

Setting up Standards

MET/CAL can set up remotely programmable and manual standards. An instrument FSC is used to set up a standard for which there is a built-in MET/CAL driver. A device for which MET/CAL has a built-in driver is referred to as a *System Instrument*.

Example, Setting up a System Instrument:

```
STEP   FSC   RANGE NOMINAL   TOLERANCE    MOD1     MOD2   3   4 CON
1.001 5700 400    350.0mV   2.9U         60H                  2W

2.001 5700        350.0mV                60H                S 2W
```

The above statements set up the 5700A to output 350mV at 60Hz with internal sense (2-wire). Statement 1.001 will also set up the UUT range and perform an evaluation. Statement 2.001 only sets up the standard (**MOD4** = **S**). Both statements will generate an automatic connection message unless otherwise inhibited using the **ASK-** FSC.

Remotely programmable standards for which there is no built-in driver are referred to as *User-configured Instruments.*

The IEEE and PORT FSCs are used to set up IEEE-488 and RS-232 controlled user-configured instruments, respectively. The IEEE and PORT FSCs must contain the correct protocol for controlling the standard, as described in the remote programming section of the instrument's operating manual.

Example, Setting up a User-configured Instrument:

```
1.001  IEEE          [@2812]:INST:NSEL 1;:VOLT 2;:CURR 1
```

The above statement sets up the PM 2812 DCV power supply channel 1 to 2V with a current limit of 1A.

Manual standards are set up by the operator under the direction of prompts generated by the DISP, MESS, and PIC FSCs.

Example, Setting up a manual standard:

```
1.001  DISP          Set the 792A to the 220mV range.
```

Setting up Memory Registers

The correct memory registers must be set up with the values of UUT Indicated and System Actual for the test evaluation to be properly performed. These registers are automatically set up by instrument FSCs that perform an evaluation (**MOD4** = *blank*) as shown in the simple 5700A example earlier in this section. However, there are many tests where an instrument FSC evaluation cannot be used. For these tests, one or more additional FSCs must be used to set up the UUT Indicated and System Actual values in the correct memory registers. This is the case for *closed-loop* tests and tests where the UUT units are not the same as the units of the applied standard.

Example, closed-loop test:

```
STEP  FSC  RANGE NOMINAL   TOLERANCE   MOD1   MOD2  3  4 CON
1.001 5700       10.0000V                            S  2W
1.002 IEEE       ?[I]
1.003 MEME
1.004 MEMC  20    V        0.0005U
```

The action of each test step in the above example is detailed below:

1. **Setup** - The 5700A is programmed to output 10V DC and a value of 10 is stored in memory register MEM1.
2. **Setup** - The UUT reading is read via the IEEE-488 bus and stored in memory register MEM.
3. **Setup** - Swap the contents of registers MEM and MEM1. This is necessary so System Actual, the 5700 output, and UUT indicated, the UUT reading, are in the correct registers for the MEMC (memory compare) evaluation FSC.
4. **Evaluation** - Compare the UUT Indicated to System Actual against the UUT specification in the Tolerance field.

Example, units of UUT parameter and applied parameter differ:

```
STEP   FSC   RANGE NOMINAL  TOLERANCE   MOD1  MOD2  3  4 CON
1.001  5700        2V                   60H           S 2W
1.002  ACC         2.828Vp  75P% 7e-6U
1.003  MEMI        Enter UUT reading in volts peak:
1.004  MEME
1.005  MEMC        Vp       0.045U
```

The action of each test step in the above example is detailed below:

1. **Setup** - The 5700A is programmed to output 2Vrms at 60Hz and store a value of 2 in memory register MEM1.
2. **Setup** - Establish the system accuracy in volts peak for the 5700A output of 2Vrms at 60Hz. 2.828 is stored in memory register MEM1, overwriting the value of 2 stored there in step 1.
3. **Setup** - Obtain the UUT reading and store it in memory register MEM.
4. **Setup** - Swap the contents of registers MEM and MEM1. This is necessary so System Actual, the 5700A output in volts peak, and UUT indicated, the UUT reading, are in the correct registers for the MEMC evaluation.
5. **Evaluation** - Compare the UUT Indicated to System Actual against the UUT specification in the Tolerance field.

A number of FSCs can be used to set up memory registers. For a complete description of these FSCs, see **ACC**, **MATH**, **MEM2**, **MEME**, **MEMI**, and **OPBR** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

The IEEE and PORT FSCs can be used to set up memory registers MEM and MEM2 by using the **[I]** and **[I$]** special constructions respectively. Also see "Use of Standard Memory Registers" in the chapter covering General Rules for FSCs in the *MET/CAL Reference Manual*.

Evaluation

Each test ends with a test evaluation. The basic result of an evaluation is that the test passed or failed. Evaluations can be either numeric or Go-NoGo. Numeric evaluations save numeric data in the results for the test. Go-NoGo evaluations only save a pass or fail indication for the test.

Numeric Evaluation

A numeric evaluation is performed by the following FSCs:

- An instrument FSC evaluation test where the standard is sourcing an output to the UUT and slew or keyboard entry mode is specified.
- An instrument FSC evaluation test where the standard is measuring an output of the UUT.

- The **MEMC** FSC.
- The **DOSE** FSC when the user-written program appends the percent error for the test to ***dosdose.dat***.

A numeric evaluation may have additional qualifiers that can be added to the evaluation result. For example, a *marginal pass* is an evaluation in which the error, expressed as a percentage of the test tolerance, is less than or equal to 100% (a configurable value), but greater than the threshold value (also configurable, but usually 70%).

Instrument FSC Evaluation Test

Instrument FSCs are FSCs that refer directly to system standards for which MET/CAL provides a built in driver. An evaluation test is specified for an instrument FSC when the MOD4 field is blank. An Instrument FSC Evaluation Test compares the System Actual to the UUT Indicated value for the test against the UUT specification in the Tolerance field.

Example, standard sourcing an output to the UUT:

```
STEP   FSC    RANGE NOMINAL    TOLERANCE   MOD1    MOD2   3   4 CON
1.001 5700   400    350.0mV    3.9U        60H                   2W
```

The above statement will use the slew mode to obtain the UUT reading unless otherwise specified. When slew mode is used, the UUT Indicated value is taken from the Nominal field and the System Actual is the final slew value. See "Slew Dialog" under *Run Time Prompts* later in this chapter.

The keyboard entry mode is used if the K-flag is set when the statement is executed or the B-flag is set and the operator chooses *keyboard* when the *Test Evaluation* dialog is displayed. The K-flag and B-flag are set using the ASK+ FSC. For keyboard entry mode, the System Actual is the Nominal field value and the UUT Indicated is the value entered from the keyboard. See *Keyboard Entry Operator Prompt* also under *Run Time Prompts.*

For a complete description of enabling and disabling keyboard entry and slew modes, refer to **ASK+** and **ASK-** in the FSC Reference chapter of the MET/CAL Reference Manual.

Example, standard measuring an output of the UUT:

```
STEP   FSC    RANGE NOMINAL    TOLERANCE    MOD1   MOD2   3   4 CON
1.001 34401 400    350.0mV    3.9U         60H                   2W
```

In this example, where the standard is measuring an output of the UUT, the Nominal field is taken as the UUT Indicated output. The value measured by the standard is used as the System Actual.

MEMC FSC

The MEMC FSC performs a numeric evaluation similar to an instrument FSC evaluation test. However, unlike an instrument FSC, MEMC cannot measure the UUT output or obtain the UUT reading via slew or keyboard entry mode. The UUT data must be obtained prior to execution of the MEMC statement. The MEMC FSC is typically used in *closed-loop* tests and tests where the UUT units are not the same as the units of the applied standard.

Example, closed-loop test:

```
STEP  FSC   RANGE NOMINAL   TOLERANCE  MOD1   MOD2  3  4 CON
1.001 5700        10.0000V                          S 2W
1.002 IEEE        ?[I]
1.003 MEME
1.004 MEMC  20    V         0.0005U
```

The action of each test step in the above example is detailed below:

1. **Setup** - The 5700A is programmed to output 10V DC and a value of 10 is stored in memory register MEM1.
2. **Setup** - The UUT reading is read via the IEEE-488 bus and stored in memory register MEM.
3. **Setup** - Swap the contents of registers MEM and MEM1. This is necessary so System Actual, (the 5700A output in volts peak), and the UUT indicated are in the correct registers for the MEMC evaluation.
4. **Evaluation** - Uses the UUT Indicated, the System Actual, and the UUT specification in the Tolerance field to determine whether the test passed or failed.

Example, units of UUT parameter and applied parameter differ:

```
STEP  FSC  RANGE NOMINAL   TOLERANCE   MOD1   MOD2  3  4 CON
1.001 5700 2V                         60H            S  2W
1.002 ACC        2.828Vp   75P% 7e-6U
1.003 MEMI       Enter UUT reading in volts peak:
1.004 MEME
1.005 MEMC       Vp        0.045U
```

The action of each test step in the above example is detailed below:

1. **Setup** - The 5700A is programmed to output 2Vrms at 60Hz and store a value of 2 in memory register MEM1.
2. **Setup** - Establish the system accuracy in volts peak for the 5700A output of 2Vrms at 60Hz. 2.828 is stored in memory register MEM1, overwriting the value of 2 stored there in step 1.
3. **Setup** - Obtain the UUT reading and store it in memory register MEM.
4. **Setup** - Swap the contents of registers MEM and MEM1. This is necessary so

the System Actual (the 5700A output in volts peak), and the UUT Indicated (the UUT reading) are in the correct registers for the MEMC evaluation.

5. **Evaluation** - Uses the UUT Indicated, the System Actual, and the UUT specification in the Tolerance field to determine whether the test passed or failed.

DOSE FSC (Numeric Evaluation)

The DOSE FSC runs a user-provided program or batch file. The DOSE FSC behaves like a numeric evaluation when the user-written program or batch file appends the percent error for the test to ***dosdose.dat***.

For a complete description of this FSC, see **DOSE** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

Go-NoGo Evaluation

A Go-NoGo evaluation is performed by the following FSCs:

- An instrument FSC evaluation test where the standard is sourcing an output to the UUT and Go-NoGo mode is specified.
- The **EVAL** FSC.
- The **PICE** FSC.
- The **DOSE** FSC when the user-written program *does not* append the percent error for the test to ***dosdose.dat***.

Instrument FSC Evaluation Test

Instrument FSCs are FSCs that refer directly to system standards for which MET/CAL provides a built in driver. An evaluation test is specified for an instrument FSC when the MOD4 field is blank. An Instrument FSC Evaluation Test uses the System Actual, the UUT Indicated value for the test, and the UUT specification in the Tolerance field to determine whether the test passed or failed.

Example, Go-NoGo Instrument Evaluation Test.

```
STEP   FSC   RANGE NOMINAL    TOLERANCE    MOD1    MOD2   3   4 CON
1.001 ASK+         G
1.002 5700 400     350.0mV    3.9U         60H                  2W
```

The above statement will use the Go-NoGo evaluation if the G-flag is set when the statement is executed or the B-flag is set and the operator *does not* choose *keyboard* or *slew* when the *Test Evaluation* dialog is displayed. The G-flag and B-flag are set using the ASK+ FSC. When Go-NoGo mode is used, no numeric data for the test is stored in the results.

For a complete description of enabling and disabling Go-NoGo mode, refer to **ASK+** and **ASK-** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

EVAL FSC

The EVAL FSC presents a message to the operator and typically waits for a YES or NO response. The operator's response determines the result of the test (PASS or FAIL). See *Test Evaluation Prompt* under *Run Time Prompts* later in this section.

Example

```
1.009  EVAL   Did the UUT display test pass?
```

For a complete description of this FSC, see **EVAL** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

PICE FSC

The PICE FSC displays a picture to the operator and waits for a YES or NO response. The operator's response determines the result of the test (PASS or FAIL). See *Pictorial Information Prompt* under *Run Time Prompts* later in this section.

For a complete description of this FSC, see **PICE** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

DOSE FSC (Go-NoGo Evaluation)

The DOSE FSC runs a user-provided program or batch file. The DOSE FSC behaves like a Go-NoGo evaluation when the user-written program or batch file appends *PASS* or *FAIL* for the test to `dosdose.dat`.

If neither PASS nor FAIL is appended, a pass condition is assumed.

For a complete description of this FSC, see **DOSE** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

Procedure Flow Control

When MET/CAL executes a procedure, it starts on the first line after the header and executes the lines in order until one of the following occur:

1. The end of the procedure is reached.
2. An **END** statement is executed.
3. A **JMP**, **JMPF**, **JMPT**, **JMPL**, or **JMPZ** statement transfers control to a different procedure statement.
4. A repeat statement (**RPT**) transfers control to Step 1 of the test.

5. A **CALL** statement transfers control to a subprocedure.
6. The operator alters the normal procedure execution sequence using the Post Test **List**, **Repeat**, **Cancel**, or **Terminate** options.

The **OPBR** (Operator Branch) FSC can be used with the JMPF, JMPT, or JMPL FSCs to perform a jump based on the operator's response to a Yes/No prompt. See *Test Decision Prompt* under *Run Time Prompts* later in this section.

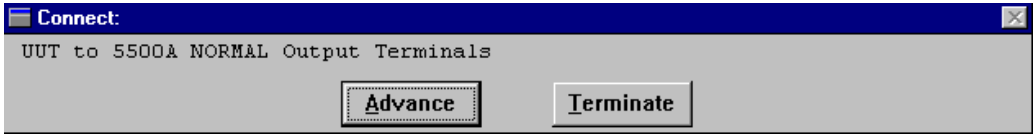
For a complete description of these FSCs, see the FSC Reference chapter of the MET/CAL Reference Manual.

Run Time Prompts

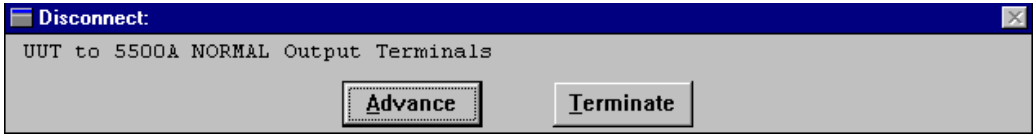
Automatic Connection and Disconnection Message Prompts

Automatic connection and disconnection message prompts are generated by instrument FSCs and the CON (connection) FSC. Automatic connection messages generally have limited utility, because MET/CAL builds them with only the information contained in the CON (UUT connection) field of an instrument FSC. For example, the following statements would generate the subsequent automatic connection and disconnection messages:

```
1.001  5500   4   3.500V     0.034U      60H     SI        2W
2.001  5500       100.0mA    1.2U                          2W
```



The automatic disconnection message prompt is displayed when the UUT connection changes. In this case, the 5500A NORMAL outputs are used for voltage, however, the AUX outputs are used for current.



Statement 1.001 specifies an AC voltage with a 2-wire connection to the UUT. This is enough information for MET/CAL to determine that the 5500A NORMAL output terminals must be used. However, this is not enough information to determine exactly which UUT terminals must be connected to the 5500A NORMAL output terminals. Typically, a procedure writer inhibits automatic connection and disconnection messages using the ASK- FSC, and instead uses a more descriptive prompt using the DISP or PIC FSCs. See the headings titled, "Operator Information Prompt" and "Pictorial Information Prompt" later in this chapter.

Example:

```
1.003  DISP      Connect the 5500A to the UUT as follows:
1.003  DISP      5500A NORMAL Output Hi to UUT V-Ohm-Diode
1.003  DISP      5500A NORMAL Output Lo to UUT Common
```

For a complete description of enabling and disabling automatic connection and disconnection messages, refer to **ASK+** and **ASK-** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

Automatic Range Message Prompt

Automatic range message prompts are generated by instrument FSCs and the RNG (range) FSCs. Automatic range messages are built by MET/CAL using the information contained in the RANGE field of the instrument FSC and the NOMINAL and MOD1 fields of the RNG FSC. For example, the following 5500 and RNG statements would generate the same automatic range message (shown immediately after the fragment):

```
13.001  5500   4     3.500V     0.056U      1kH     SI    2W
2.001   RNG          4V                     AC
```

For more information on UUT range messages, see the following sections in the *MET/CAL Reference Manual*:

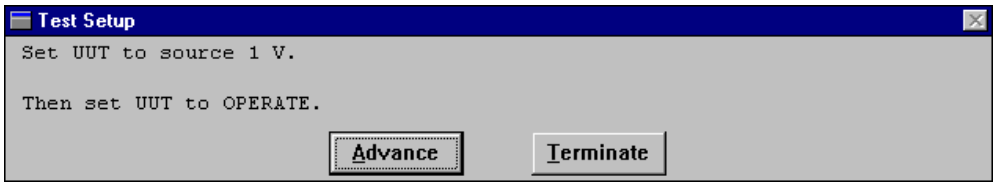
- RANGE Field under “Instrument FSCs” in General Rules for FSCs
- UUT Range Message under “Automatic Messages”, also in General Rules for FSCs
- **RNG** in the FSC Reference.

For a complete description of enabling and disabling automatic range messages refer to **ASK+** and **ASK-** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

Automatic Test Setup Prompts

Automatic test setup prompts are generated by an instrument FSC evaluation test, nominal setup test, or comparison test that is measuring an output of the UUT. For example the following 34401 statement would generate the subsequent automatic test setup prompts:

```
1.002  34401        1.00V                              N  2W
```



After the 34401A takes the measurement, the following automatic test setup prompt is generated:

For a complete description of enabling and disabling automatic test setup prompts refer to **ASK+** and **ASK-** in the *MET/CAL Reference Manual*.

Automatic UUT Setup Prompt

An automatic UUT setup prompt is generated by an instrument FSC setup test that is monitoring an output of the UUT. For example the following 34401 statement would generate the following automatic UUT setup prompt:

```
1.002  34401        1.00V                              S  2W
```

For a complete description of enabling and disabling the automatic test setup prompt, refer to **ASK+** and **ASK-** in the FSC Reference chapter of the MET/CAL Reference Manual. For an explanation of how to create a message to annotate or replace the automatic UUT setup message, see **MESS** in the same chapter.

Slew Dialog

The slew dialog is a popup screen that is created by an instrument FSC evaluation test, nominal setup test, or comparison test when the calibration instrument is applying an output to the UUT. The slew dialog is the default mode for evaluation tests and comparison tests. The slew dialog is always displayed for nominal setup tests. When the slew dialog is displayed, it is used to perform one of the two functions enumerated below:

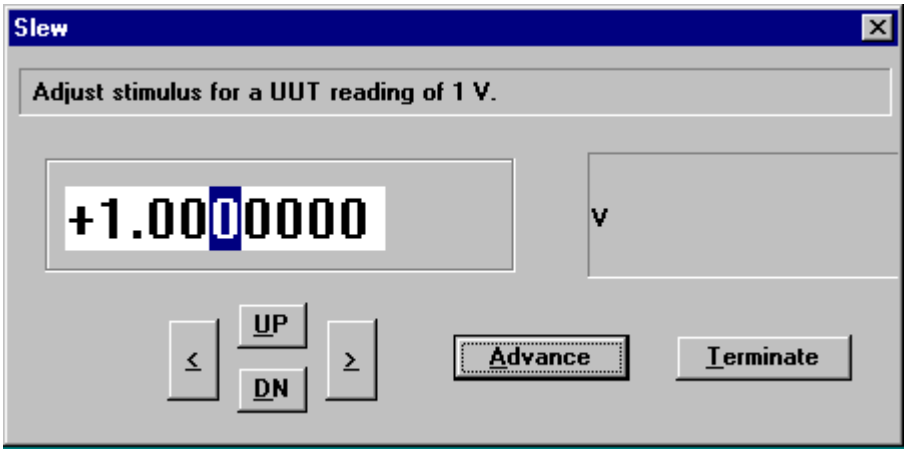
1. Slewing the Calibrator Output

If the calibration instrument allows incremental changes of its output around the nominal value, then, as the operator changes the displayed value of the slew screen, the number shown reflects the actual calibrator output. The operator slews

(adjusts) the display until the UUT is reading the nominal value as closely as possible. If the statement is an evaluation test or comparison test, MET/CAL can then calculate the deviation from the nominal required to make the UUT read the desired value and report the error of the UUT.

Example:

```
1.002  5700      1.00V          0.01U                    2W
```

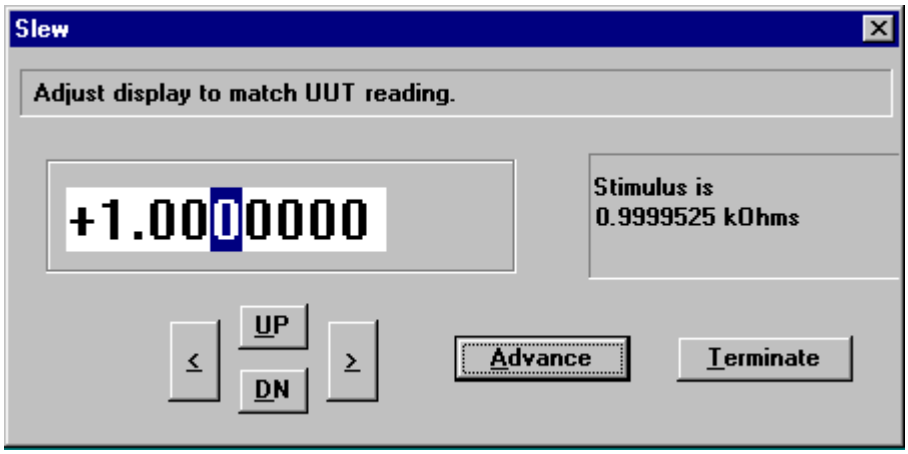


2. Matching the UUT Display

If the calibration instrument cannot incrementally change its value, like 5700A resistance, slew acts differently. The operator adjusts the displayed value in the slew screen to match the reading shown on the UUT's display. MET/CAL uses this information to calculate the UUT error.

Example:

```
1.002  5700     1.00kZ       0.01U                    4W
```



For more information on the slew dialog, see the following sections in the *MET/CAL Reference Manual*:

- Stimulus Function under "Evaluation Test"
- Stimulus Function under "Nominal Setup Test"

For information on configuring the slew message, see the following sections in the *MET/CAL Reference Manual*:

- Slew Message under "Automatic Messages" in General Rules for FSCs
- **MESS** in the FSC Reference.

Keyboard entry or Go-NoGo mode can be selected in lieu of slew mode using the appropriate ASK flags (see Keyboard Entry Operator Prompt and Test Evaluation Prompt in this section).

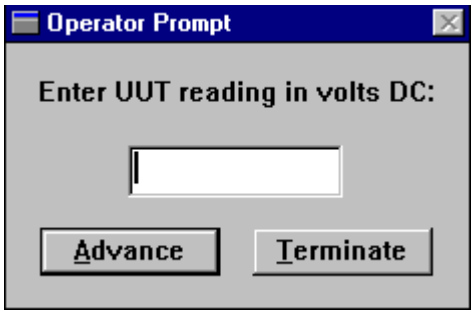
For a complete description of enabling and disabling slew mode refer to the **ASK+** and **ASK-** FSCs in the *MET/CAL Reference Manual*.

Keyboard Entry Operator Prompt

The keyboard entry operator prompt is a popup screen that is created by an instrument FSC evaluation test, nominal setup test, or comparison test to enter the UUT reading. The operator prompt is used when the keyboard entry evaluation mode is selected. The following is an example instrument evaluation test with keyboard entry mode selected:

```
1.002  ASK+          K
1.003  5700          1.00V          0.01U                    2W
```

The ASK+ K statement specifies keyboard entry mode for the 5700 evaluation test. This causes the following dialog to be displayed, prompting the operator for the UUT reading:



For more information about the Keyboard Entry Operator prompt, see *Instrument FSCs* in the chapter "General Rules for FSCs" of the *MET/CAL Reference Manual*.

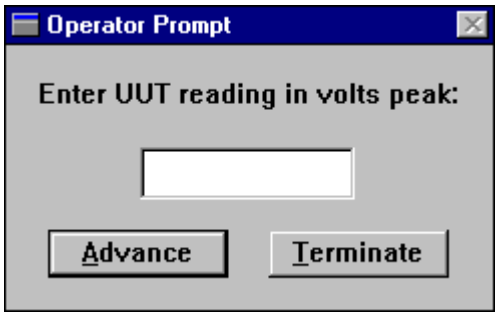
Memory Register Operator Prompts

Memory register prompts are popup screens that prompt the operator to enter data for the MEM and MEM2 registers. The MEMI FSC is used to display a prompt for the value of MEM. The MEM2 FSC is used to display a prompt for the value of the MEM2 register.

The MEMI FSC can be used to prompt for the UUT reading, like the keyboard entry prompt does for instrument FSC. Specifically, the MEMI FSC must be used to obtain the UUT reading when its units are not the same as those of the standard. This situation is illustrated in the following procedure fragment:

```
43.003  5700   2V                    60H              S  2W
43.004  DISP   Press the MIN MAX button.
43.004  DISP   Press BEEPER button to display maximum values.
43.005  ACC    2.828Vp   75P% 7e-6U
43.006  MEMI   Enter UUT reading in volts peak:
43.007  MEME
43.008  MEMC   Vp        0.045U
```

Note that the UUT reading is in volts peak even though the setup of the standard is specified in volts RMS. If a 5700 evaluation test had been used instead of a 5700 setup test, the UUT reading would have been incorrectly entered in volts RMS. The following shows the MEMI prompt that would be generated by the above procedure fragment:

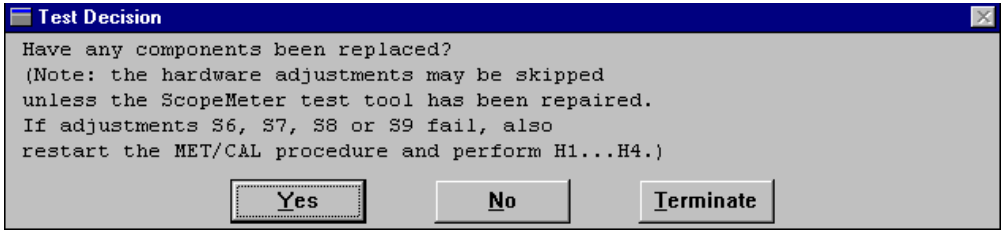


For a complete description of creating memory register prompts, see **MEMI** and **MEM2** in the FSC Reference chapter of the *MET/CAL Reference Manual*. Descriptions of the other FSCs used in the above example can also be found in the FSC Reference.

Test Decision Prompt

The test decision prompt is a popup screen that is created by the OPBR FSC. The test decision prompt is used for yes/no conditional prompts. The OPBR FSC does not perform an evaluation. The following example OPBR statements create the example dialog immediately following:

```
1.001  OPBR    Have any components been replaced?
1.001  OPBR    (Note: the hardware adjustments may be skipped
1.001  OPBR    unless the ScopeMeter test tool has been repaired.
1.001  OPBR    If adjustments S6, S7, S8 or S9 fail, also
1.001  OPBR    restart the MET/CAL procedure and perform H1..H4.)
1.002  JMPT    1.003
```



For a complete description of creating the test decision prompt, see **OPBR** in the FSC Reference chapter of the *MET/CAL Reference Manual*. A Description of the **JMPT** FSC can also be found in the FSC Reference.

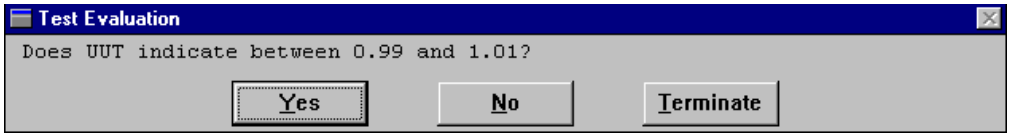
Test Evaluation Prompt

The test evaluation prompt is a popup screen that is created by an instrument FSC evaluation test or the EVAL FSC. In the case of an instrument FSC, the test evaluation prompt is generated only when the Go-NoGo evaluation of the UUT reading is selected. The following is an example instrument evaluation test with

Go-NoGo mode selected:

```
1.002  ASK+        G
1.003  5700        1.00V        0.01U                          2W
```

The ASK+ G statement specifies Go-NoGo mode for the 5700 evaluation Test.



Following is another example instrument evaluation test with Go-NoGo mode selected:

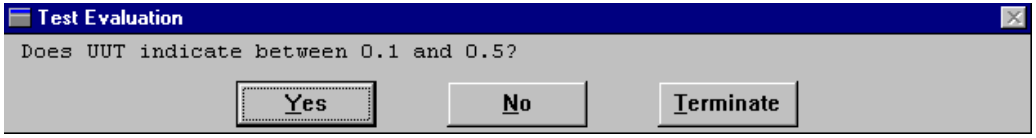
```
1.002  ASK+          B
1.003  5700          1.00V          0.01U                      2W
```

The ASK+ B statement also specifies Go-NoGo evaluation of the UUT reading, but allows the operator to optionally select either keyboard entry or slew evaluation mode.

The EVAL FSC is used for Go-NoGo evaluations that cannot otherwise be performed using an instrument FSC Go-NoGo evaluation.

Example EVAL FSC:

```
1.002  DISP  Plug a test lead into the V Ohm Diode terminal,
1.002  DISP  and touch the probe to the 10A terminal.
1.003  EVAL  Does UUT indicate between 0.1 and 0.5?
```

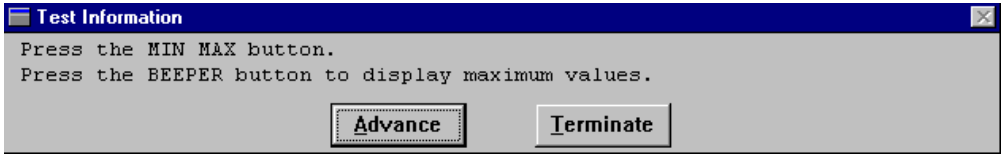


For more information about the test evaluation prompt, see *Instrument FSCs* in *General Rules for FSCs* and **EVAL** and **MESS** in the FSC Reference chapter of the *MET/CAL Reference Manual*. A Description of the DISP FSC can also be found in the FSC Reference.

Test Information Prompt

The test information prompt is a popup screen created by the DISP FSC. The test information prompt is used to present the operator with textual information that does not require a Yes or No response.

```
43.004  DISP   Press the MIN MAX button.
43.004  DISP   Press the BEEPER button to display maximum values.
```



For a complete description of creating the test information prompt, see **DISP** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

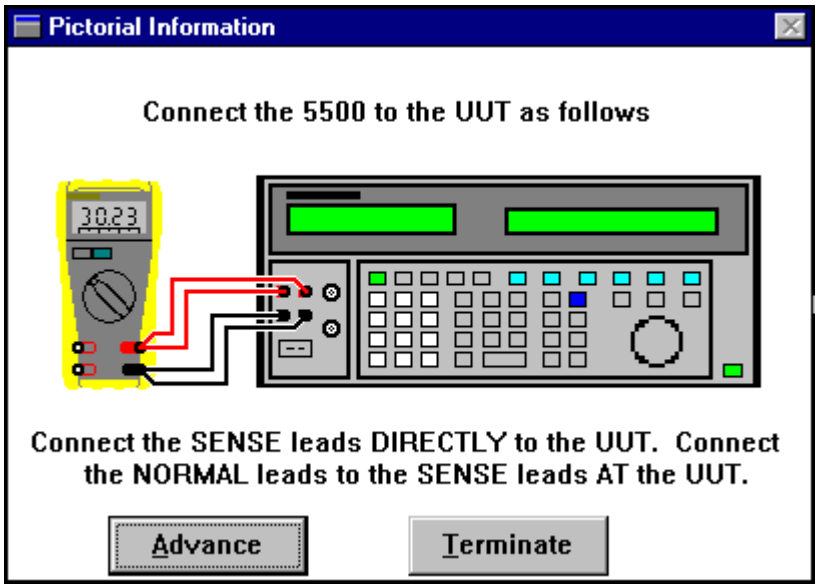
Pictorial Information Prompt

The pictorial information prompt is a popup screen that contains a picture and is created by the PIC and PICE FSCs. Pictorial information is used instead of text. The pictorial-style prompt is normally only used in those cases when instructions to the operator would require a detailed description of a required calibration step that would be difficult to present in text-only form.

The PIC FSC is used for connections and setups, whereas the PICE FSC also performs a test evaluation.

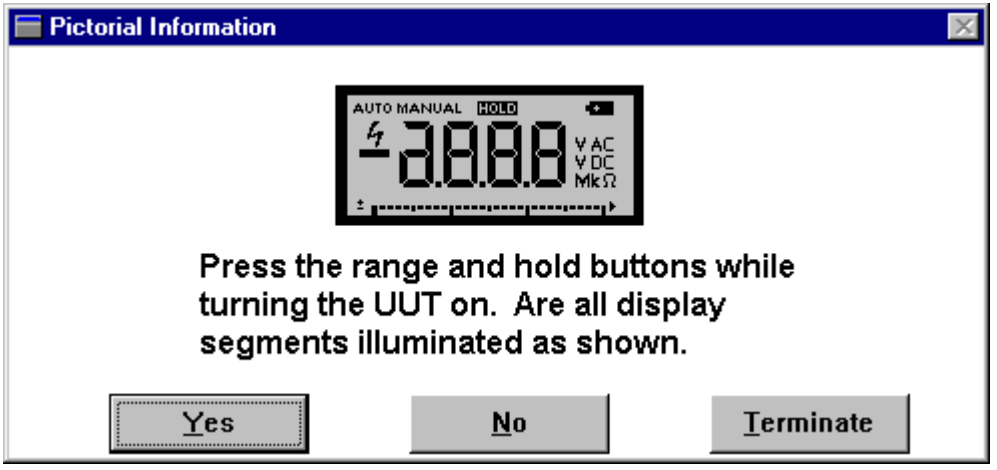
Example PIC FSC:

```
2.003  PIC          55_s3_cw
```



Example PICE FSC:

```
1.007  PICE          7x-3_dsp
```



zv405s.bmp

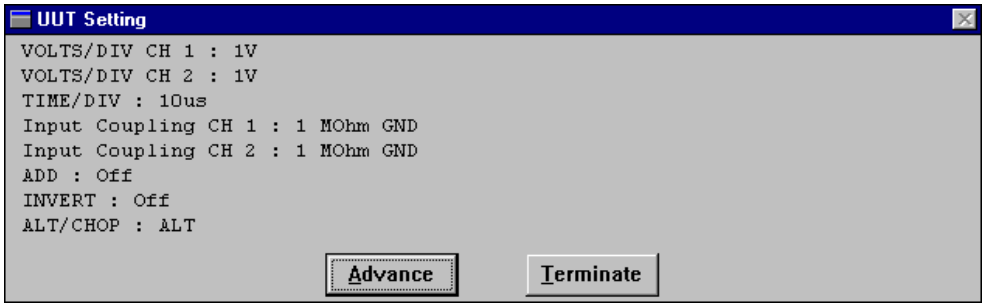
For a complete description of creating the pictorial information prompt, see **PIC** and **PICE** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

UUT Setting Prompt

The UUT setting prompt is a popup screen that contains a list of UUT setups. The prompt is generated by the SET FSC and is typically used in procedures for oscilloscopes when a large number of setups is required, or the steps are unusually complex.

Example:

```
1.001  SET          VOLTS/DIV CH 1 : 1V
1.001  SET          VOLTS/DIV CH 2 : 1V
1.001  SET          TIME/DIV : 10us
1.001  SET          Input Coupling CH 1 : 1 MOhm GND
1.001  SET          Input Coupling CH 2 : 1 MOhm GND
1.001  SET          ADD : Off
1.001  SET          INVERT : Off
1.001  SET          ALT/CHOP : ALT
```



For a complete description of creating the UUT setting prompt, see **SET** in the FSC Reference chapter of the *MET/CAL Reference Manual*.

Post Test Module

When a MET/CAL procedure executes, a series of calibration tests is performed. As described earlier, each calibration test is made up of connection messages, setup steps, and an evaluation step. The evaluation step is always the last part of a test.

The Run Time application calls the Post Test module after each evaluation step. It also calls the Post Test Module after a system error, whether or not the error occurred in an evaluation step.

The Post Test module has three main functions:

1. It resets the system instruments.
2. It generates results for the test that has just completed.
3. It displays the Post Test dialog.

System Instrument Reset

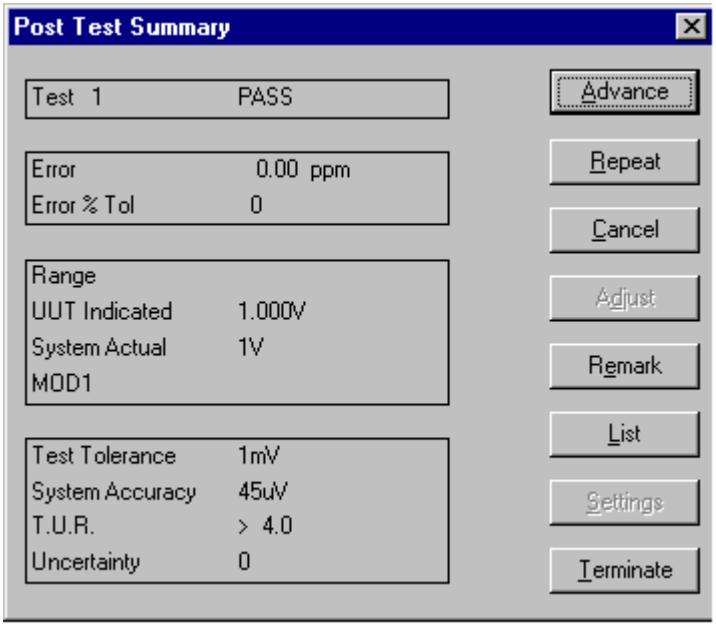
Whenever the Post Test module is called, whether because an error occurred, or because an evaluation step has been completed, one of the first things it does is reset the system instruments that are in use.

A system instrument is in use if the main procedure that is being executed, or any subprocedure it has called or may call, contains a reference to that system instrument. A reference is a use of an FSC that invokes the built-in driver for that instrument. For example, the Fluke 5700A is used by procedures that contain "5700" FSC statements. The statement need not have been executed in order for the instrument to be "in use."

It is possible to inhibit the normal resetting of the system instruments. To do this, clear the ASK 'Q' flag. Certain other ASK flags must be also clear (not set) in order to inhibit resetting of system instruments. Refer to the Reference Manual sections on the **ASK+** and **ASK-** FSCs for details.

Post Test Dialog

This section describes the Post Test Dialog as shown in the following figure.



zv401s.tif

On the left-hand side, the Post-Test Summary dialog shows the following information:

Test Field

The test field shows the test number and the test result. The test number is the major step number of the test. For example, test number 7 might look like the following:

```
7.001 DISP  Connect...
7.002 5700  1V     1%
```

The test result is shown on the right-hand side of the Test field, and will be one of the following values:

- **PASS**
  A test result value of PASS indicates that the UUT is within the test tolerance at this test point, and is also within the adjustment threshold.

  The adjustment threshold is specified in the header of the main procedure being executed. The default value is 70%. If the error, expressed as a percentage of the tolerance, is less than the adjustment threshold the test result will be PASS.

- **MARGINAL PASS**
  MARGINAL PASS indicates that the UUT is within the test tolerance at this test point, but not within the adjustment threshold.

  At some sites, the standard policy is to run the calibration adjustment procedure for a UUT if the UUT passes the verification procedure, but one or more test points generate a MARGINAL PASS result.

- **FAIL**
  FAIL indicates that the UUT is not within the test tolerance at the current test point, but is within the "significantly out of tolerance limit." See the description of this limit in the FAIL NOTIFY USER section below.

  The [startup] section of the MET/CAL initialization file `metcal.ini` contains a parameter named *pass100* that allows the system to be configured with respect to whether or not a test is a PASS or a FAIL when the error is exactly equal to 100% of the test tolerance.

  If pass100 = *yes* in `metcal.ini`, then the test result is MARGINAL PASS if the error is exactly equal to the test tolerance.

  If pass100 = *no* in `metcal.ini`, then the test result is FAIL if the error is exactly equal to the test tolerance.

  The default value of *pass100* (as distributed) is *yes*.

- **FAIL NOTIFY USER**
  FAIL NOTIFY USER indicates that the UUT is not within the test tolerance at the current test point, and also not within the "significantly out of tolerance limit."

  This limit is specified using the *oot_lim* parameter in the [startup] section of the MET/CAL initialization file `metcal.ini`. Legal values for the *oot_lim* parameter are between 99.9 and 500.0.

  The purpose of the *oot_lim* (out of tolerance limit) parameter is to allow a site to specify a limit above which a failing UUT is considered to be significantly out of tolerance, potentially requiring the recall of instruments recently calibrated or adjusted using the UUT in question.

  The default value of *oot_lim* (as MET/CAL is distributed) is 100.0. This is a conservative setting, indicating that any UUT that fails at any test point will generate the FAIL NOTIFY USER test result.

Error Field
The error is calculated as:

error = UUT Indicated – System Actual

If the NOMINAL units are dBm, the Nominal value is zero, the UUT Indicated value is zero, or the System Actual value is zero, the error is formatted as an absolute number.

Otherwise, the error is formatted as a percentage of either the UUT Indicated or the NOMINAL value, depending on the value of the *tol_ref_nominal* parameter in the MET/CAL initialization file.

For a Go/NoGo test the Error field is blank.

Error % Tol Field
For a numeric evaluation, this field shows the error expressed as a percentage of the test tolerance.

For a Go/NoGo evaluation the *Error % Tol* field is blank. This applies to instrument evaluations with ASK+ 'G' set, to EVAL statements, PICE statements, and to some DOSE statements.

Range Field
The Range field shows the UUT range, as indicated in the RANGE field of the procedure statement just executed. The Range field shows "A" to indicate auto-range.

UUT Indicated Field
The UUT Indicated field shows the UUT Indicated value for the test. For a test in which the calibration standard is a stimulus device (like a Fluke 5700A), and the UUT is a measurement device (such as a Fluke 77), the UUT Indicated value is the reading taken by the UUT. If the test is a Go/NoGo test, the UUT Indicated field is blank on the Post Test dialog.

For a test in which the calibration standard is a measurement device (such as an HP 3458A), and the UUT is a stimulus device, the UUT Indicated value is the expected reading, as indicated in the NOMINAL field of the procedure statement that performed the evaluation step.

System Actual Field
The System Actual field shows the System Actual value for the test.

If the UUT is a measurement device, the System Actual is the true value applied to the UUT by the calibrator being used as a calibration standard.

If the UUT is a stimulus device, the System Actual is the measurement taken by the measurement device being used as the calibration standard.

MOD1 Field
The Post Test MOD1 field shows the information in the MOD1 field of the procedure evaluation step just executed. If you execute an evaluation step such as the following:

```
STEP NOMINAL MOD1 TOLERANCE
5700 10kH    1V   1%
```

the evaluation is based on frequency (NOMINAL field), so you will see that MOD1 shows the voltage (1V in this example).

Test Tolerance Field
The Test Tolerance field shows the test tolerance in units of the NOMINAL field.

For example, suppose you execute the procedure evaluation step:

```
5700 100V          1%
```

The test tolerance displayed in the Post Test dialog will be 1V, because 1V is 1% of 100V.

System Accuracy Field
The System Accuracy field shows the accuracy of the system instrument (calibration standard) used for the test.

For example, suppose you execute the following procedure evaluation step:

```
5700 200V           1%
```

If the 90-day accuracy file for the 5700A is used, the system accuracy will be shown as 1.3mV.

This is because in the accuracy file, for the 220V range, the tolerance is given as 6e-4 and the floor is 100e-6V. Note that the tolerance value in the accuracy file is a percentage.

The calculation is therefore:

system accuracy = 6e-4 * .01 * 200V + 100e-6V

... which is equal to 1.3mV, as shown in the Post Test dialog.

If the test in the procedure is a MEMC test in which the system accuracy is specified using an **ACC** statement, MET/CAL uses the value from the ACC statement and does not look up the system accuracy in one of the FSC-specific system accuracy files.

T.U.R. Field
The T.U.R. field shows the test uncertainty ratio for the test.

The test uncertainty ratio is the ratio of the test tolerance to the system accuracy.

If the test tolerance or the system accuracy are specified in decibels (dBm), the quantity is converted to volts before it is used to calculate the T.U.R.

If the T.U.R. exceeds the T.U.R. limit specified in the MET/CAL initialization file, the Post Test dialog does not show the T.U.R. Rather, the dialog indicates that that the T.U.R. is greater than the limit, and shows the limit.

The T.U.R. limit is specified by the *tur_lim* parameter in the [startup] section of the MET/CAL initialization file, `metcal.ini`. The default value for the limit is 4.0. Any value between 0 and 10 can be specified. It is also legal to set *tur_lim* to *none*, which disables the calculation of the T.U.R. by the Run Time and Editor applications. If *tur_lim* is *none*, the Post Test dialog does not show values in the Test Tolerance, System Accuracy, or T.U.R. fields.

Post Test Summary Dialog Buttons

On the right-hand side of the Post-Test Summary dialog are 8 buttons. These have the following functions:

Advance
Choosing **Advance** causes procedure execution to resume at the next step.

Repeat
The basic function of **Repeat** is to repeat the current step. The precise operation of Repeat depends on the current setting of the ASK 'P' flag in the procedure.

If the 'P' flag is set, the procedure is re-started at the current procedure line when the operator selects the Post Test Repeat option.

If the 'P' flag is *not* set, the procedure is re-started at the first statement of the current test. The step number of the first statement of the test has the form XX.001.

Note that when a test is repeated more than once (that is, executed more than twice), MET/CAL saves only the first and last results for that test step.

Cancel
Selecting **Cancel** is the same as Repeat (see above) except that when the operator chooses Cancel, the result of the canceled test is not saved.

As with Repeat, the procedure step on which execution resumes is controlled by the current setting of the ASK 'P' flag.

The Post Test Cancel button can be disabled (grayed out) by not setting the ASK 'X' flag in the procedure.

A parameter *ask_x_default* in the [startup] section of the MET/CAL initialization file `metcal.ini` controls the default state of the ASK 'X' flag.

- If *ask_x_default* is set to *yes* (or +), then the ASK 'X' flag will be set by default in a procedure that does not specify otherwise. This means that the Post Test Cancel button is enabled by default.

- If *ask_x_default* is set to *no* (or –), then the ASK 'X' flag will not be set by default in a procedure that does not specify otherwise. This means that the Post Test Cancel button is disabled by default.

  As distributed, *ask_x_default* is set to *yes* in `metcal.ini`.

Adjust
The **Adjust** button is enabled only if the error for the test just completed exceeds the adjustment threshold, and the procedure being executed contains an adjustment block for the current test step, and the ASK 'J' flag is set.

There is some interaction between the ASK 'J', 'A', and 'F' flags. Refer to the description of the "ASK+" and "ASK-" FSCs in the MET/CAL *Reference Manual* for details.

Procedures distributed by Fluke do not make use of in-line adjustment blocks. Separate verification and adjustment procedures are provided. When these procedures are used, the Adjust button will never be enabled.

Remark
Choosing **Remark** allows a remark of up to 32-characters to be entered. The entered remark is written to the results. Multiple remarks can be entered for one test step.

List
Choosing **List** causes a read-only listing of the current procedure to be displayed. When the listing is displayed, the current line (that is, the line just executed) is highlighted.

The highlighted line can be changed using the Up Arrow, Down Arrow, Page Up, and Page Down keys, or by using the mouse to click on the desired line.

Note that when the Run Time first displays the listing, the input focus is on the Advance button. The Up Arrow, Down Arrow, Page Up, and Page Down keys will not work as expected until the focus has been moved to the list box that shows the procedure. This can be done using the Tab key, or by clicking on the list box.

The Procedure Listing window includes Advance and Terminate buttons. These are not the same as the Advance and Terminate buttons in the Post-Test Summary dialog. Choosing **Advance** in the Procedure Listing causes procedure execution to resume by executing the highlighted line.

It is important to realize that if you choose the Post Test **List** button, and then immediately choose **Advance** in the Procedure Listing window, you will re-execute the step that just completed. If you really just wanted to look at the procedure, and would like to resume execution as if you hadn't chosen List at all, you can choose **Terminate** in the Procedure Listing window, followed by **Advance** in the Post-Test Summary dialog, or you can move the highlighted line down one line in the Procedure Listing window, and then choose **Advance** there.

The ASK 'L' flag can be used to enable or disable the List option in the Post-Test Summary dialog.

Settings
Choosing the **Settings** button causes the Run Time to display the last 18 settings created by the **SET** FSC. This allows the operator to verify that the UUT is set up correctly for the current test. This capability is used primarily in conjunction with calibration oscilloscopes. Refer to the SET FSC section in the *MET/CAL Reference Manua*l for details.

The Settings button is disabled unless one or more stored settings are available to be viewed.

Terminate
Choosing **Terminate** stops the execution of the procedure. Results generated before the procedure is terminated are not discarded, unless the operator subsequently chooses to discard them.

The Terminate button can be enabled or disabled by setting or clearing, respectively, the ASK 'T' flag. By disabling the Terminate and List options, a procedure writer can write a procedure that, barring errors, forces the operator to execute all of the tests in the procedure. Such a procedure should be carefully tested, however, because it can be difficult to terminate.

Test Results Window

As mentioned in the Run Time outline at the beginning of this chapter, the Test Results window is a separate window that shows a one-line, color-coded result for each test step in the Calibration Procedure.

The Test Result window initially appears in minimized form. It can be opened, moved, re-sized, or minimized at any time during procedure execution.

The contents of the Test Results window are not cleared until a subsequent Calibration Procedure is started. That is, after you finish a Calibration Procedure and return to the top-level Run Time menu, you can still look at the results of the procedure in the Test Results window. However, as soon as you start another procedure, or start to re-execute the same procedure, the Test Results window is cleared.

The lines in the Test Results window are color coded as follows:

- GREEN indicates the test result is PASS.
- YELLOW indicates the test result is MARGINAL PASS.
- RED indicates the test result is FAIL.

A **PASS** is a test in which the error, expressed as a percentage of the test tolerance, is less than or equal to the adjustment threshold.

A **MARGINAL PASS** is normally a test where the error, expressed as a percentage of the tolerance, is greater than the adjustment threshold, and less than or equal to 100%. The adjustment threshold is set in the header of the main procedure and is, by default, 70%.

A **FAIL** is normally a test in which the error, expressed as a percentage of the test tolerance, is greater than 100%.

The reason the above descriptions of MARGINAL PASS and FAIL say "normally" is that MET/CAL is configurable with respect to whether a test at exactly 100% of tolerance is considered a MARGINAL PASS or a FAIL.

If the parameter *pass100* in the [startup] section of the MET/CAL initialization file `metcal.ini` is set to *yes*, then exactly 100% of tolerance is regarded as a MARGINAL PASS, and the error must be strictly greater than 100% in order for the test to be a FAIL. If pass100 is set to *no*, then exactly 100% of tolerance is a FAIL, and the error must be strictly less than 100% in order for the test to be considered a PASS or MARGINAL PASS.

However, note that there may be cases when the value displayed in the %Tol field does not appear to agree with the PASS/FAIL result of the test. This happens when the actual error, expressed as a percentage of the tolerance, is rounded up or down and displayed as 100%. For example, if the error as a percentage of the tolerance is 99.9999%, it will be displayed as 100%, but the test will be a PASS (actually a MARGINAL PASS), even if pass100 is set to *no*. This consideration is not specific to the Test Results window. It also applies to the Post Test dialog (discussed elsewhere in this chapter), and to the formatted results that MET/CAL writes to the database.

If the operator terminates a test manually, the Test Results window displays a one-line message in red indicating that the statement was terminated by the operator.

The Test Result window can display a maximum of about 900 result lines from a single procedure run. If this limit is exceeded, the display functions as a first-in/first-out queue, always showing approximately the last 900 result lines.

The content and format of the lines displayed in the Test Results window are fixed and cannot be customized by the user.

For a normal instrument evaluation test, the Test Results window shows:

No..... test step number

Range..... UUT range (if specified)

UUT Indicated.... value indicated by UUT for test

System Actual..... actual stimulus or measured value for test

C..... Y if System Actual correct using a correction file, otherwise N

Modifier ..... MOD1 value from test, if any, -- often shows frequency

Error .....absolute error (UUT Indicated - System Actual)

%Tol .....error as a percentage of test tolerance

For evaluation steps based on EVAL, PICE, DOSE statements, or instrument Go/No Go evaluation steps, some of the information is not available, and the field will be blank in the Test Results window.

For example, in a Go / No Go test the operator is simply asked if the UUT Indicated value is between a specified set of bounds. The answer is Yes or No, and does not indicate a numerical error. Because of this you will see OPER-EVAL in the UUT Indicated column of the Test Results window, and the Error and %Tol fields will be blank.

Safety Window

The safety window is an always-on-top popup display that the Run Time creates whenever a potentially dangerous voltage is being generated during procedure execution by one of the configured system instruments the procedure is using.

The word **Danger** is shown in the window title, and the standard lightning bolt danger symbol is shown below the title.

The safety window does not appear when a dangerous voltage is generated by a user-configured instrument, or by any instrument being controlled by the IEEE or PORT FSCs.

A dangerous voltage is any DC voltage greater than 60V, or any AC voltage greater than 42.4V (peak).

Operators are cautioned not to rely only on MET/CAL's safety window as an indication of a dangerous voltage. There are two reasons for this:

- First, instruments not under control by a built-in MET/CAL instrument driver do not cause the safety window to appear when a dangerous voltage is present.
- Second, it is possible to manually override the setting of a calibrator so that it is generating an unsafe voltage without the Run Time's knowledge.

Always verify that instruments are in standby before touching the output terminals.

The Run Time application attempts to disable the operation of screen savers during procedure execution in order to prevent the safety window from being covered up by the screen saver display. However, some screen savers do not conform to the guidelines for screen savers, so it can happen that the safety window will be covered up by a screen saver.

Users are advised to disable screen savers when using the MET/CAL Run Time application to calibrate instruments.

Post Prompt Dialog

When a calibration is complete or the procedure is terminated, the Run Time displays the Post Prompt Dialog.

You can use the Post Prompt dialog to add administrative information necessary for the results. After adding the information, click **OK**. The following figure indicates typical information you can add to the results. Information entered on the Post Prompt Dialog can be included in the final calibration report.

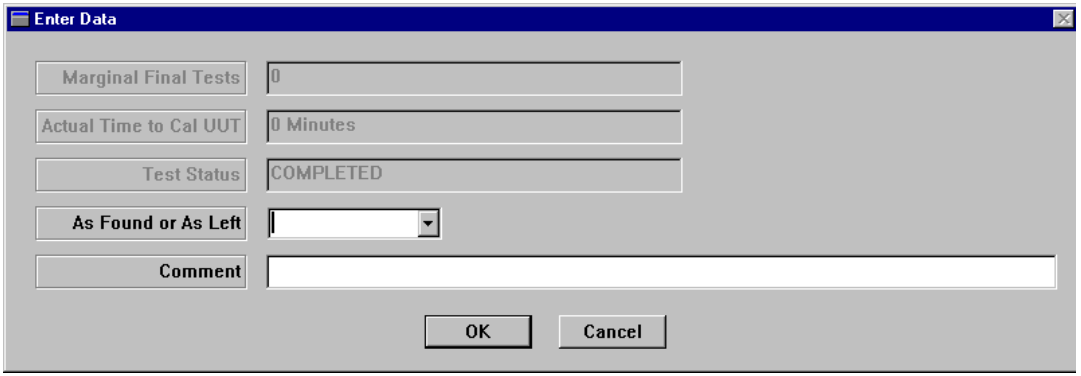


Figure 4-3. Sample Post Prompt Dialog

Note

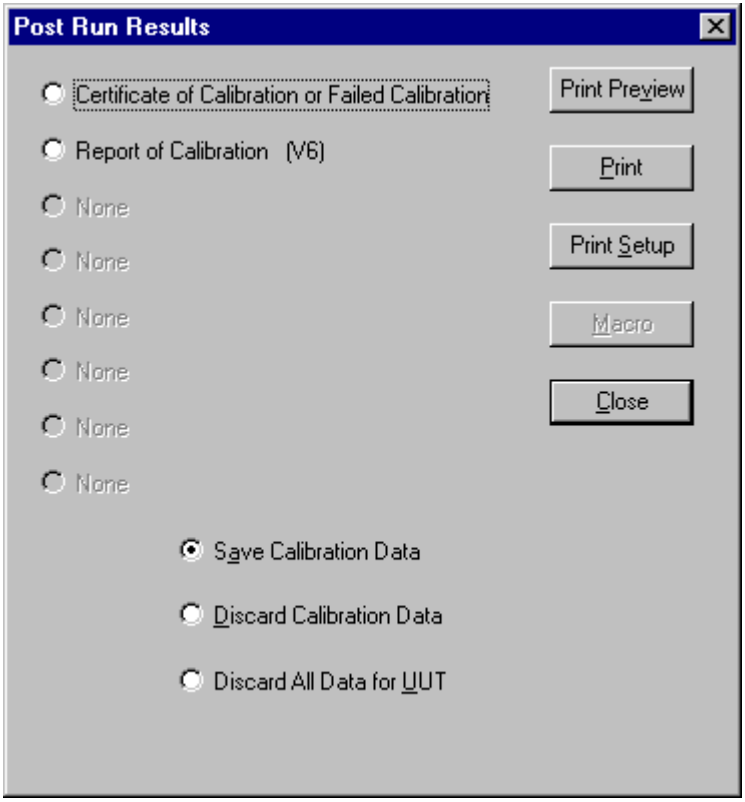
The fields in the form may or may not be filled automatically, depending on how you have configured the PASS_DB.TXT and FAIL_DB.TXT files.

Post Run Results

After you close the Post Prompt dialog, the Run Time displays the Post Run Results dialog, as shown in the following figure.

Note

The following section describes all possible items in the Post Run Results dialog. However, using the Fluke Metrology Software Configure menu under the Run Time application, you can configure the dialog to exclude use of some of the items or to inhibit the dialog box from opening. Also, you can customize which reports are available.



zv402s.tif

Figure 4-4. Post Run Results Dialog

Post Run Reports

The dialog can be configured to permit selection of up to eight different reports. The configuration of the software as shipped has three of the possible eight reports already selected. These reports are as follows:

- **Certificate of Calibration #1**: a calibration certificate that lists the standards used to calibrate the UUT, the environmental conditions during calibration, who signed it, and when they signed it.
- **Calibration Results Report #1:** a test results report with detailed test results for each step.
- **Calibration Summary Report #1**: a summary report that summarizes the full results report.

Print Preview

With one of the choices selected, click **Print Preview** to view the certificate or report without actually printing it. Use this selection to identify the report you want and to verify the information it contains.

Print

The Print selection allows you to print calibration results, summaries, and certificates. To print, proceed as follows:

1. Select one of the reports.

   Click **Print**. The print dialog box opens. If the settings are not correct, make the desired changes in the settings.

2. Click **OK** when you are satisfied with the settings (otherwise **Cancel**).

   The Run Time sends the certificate or report to the printer and returns you to the Post Run Results dialog.

Print Setup

Use Print Setup to select the printer, printing orientation, and other options.

Macro

Macros provide an automated method of storing a series of actions for later replay. This lets you minimize the number of selections you make after each Calibration Procedure. You need to select only the **Macro** button on the Post Run Results dialog to execute all of the actions previously set up in the Run Time application. (**Configure** | **PostRun Macro** selection).

For example, you can use the Run Time to set up the macro to preview or print several types of reports automatically. Clicking **Macro** initiates the action.

Calibration Data on Close

You have several options for how you choose to handle calibration data when you close the application. The options are represented the following buttons:

Save Calibration Data
Test results are saved in the database when you close the dialog box.

Discard Calibration Data
Test results are not saved in the database when you close the dialog box.

Discard All Data for UUT

Test results are not saved in the database when you close the dialog box. In addition, all date for the asset (UUT) is discarded from the database when you close the dialog box. This option will fail if the asset in question has prior calibration history in the database. If this is the case, an error message is generated and the user must select one of the other two choices.

These buttons are mutually exclusive and the resulting action is permanent. The **Configure PostRun** menu in the Run Time application allows you to control availability of these options by selectively enabling or disabling them.

Chapter 5
Reports

Contents **Page**

Note

REGARDING SAMPLE REPORTS:

Fluke Metrology Software is supplied with a series of Sample Reports that support the general usage philosophy of the product. It is impossible for Fluke to provide reports that will satisfy every situation. Therefore, the sample reports are provided as examples that you can modify as your requirements dictate. If these reports are used unmodified, as supplied by Fluke, it is your responsibility to verify that they provide the output that your circumstances require. They were created and can be modified with Crystal Reports Professional, which is supplied with the base software package.

Section 1: Introduction

A fundamental reason for collecting information in a database is so you can retrieve it later and view trends and patterns within the data. Fluke MET/BASE applications give you a number of ways to see your data, from viewing it on the display to printing out multi-page reports.

Fluke Metrology Software comes with several calibration reports. These reports are designed to satisfy the needs of a typical calibration laboratory. You can easily accomplish minor changes by editing one of the supplied reports.

This chapter introduces the report process, and then demonstrates how to modify one of the supplied reports. Later sections show how to create a new report from scratch.

Types of Reports

MET/BASE uses reports that are written using Crystal Reports. You can run reports off stored procedures (explained later) or access data tables directly. Once you've written a report, it is installed into the system. This allows users to select them from a list and execute them. The way you install reports determines how they can be used in MET/BASE.

General Reports Many reports fit into this category. The supplied reports that print database customization and calibration recall lists are examples of general reports. General reports typically provide information about more than one asset or history record (i.e. the recall report). They may or may not have user prompts. If general reports have user prompts, then Crystal Report prompt should be used. In contrast to other report types, no prompts are generated for you. You can run General reports from the Manual Entry Application or from Crystal Reports.

Run Time Reports Run Time or Calibration reports are designed to print results of a specific calibration event. You can run them at any time from the Manual Entry or the Run Time Applications. Run Time reports are written around a CTAG value. In the Manual Entry application, the system will automatically prompt the user for the asset number, then for a calibration event. From this information, a specific CTAG value can be passed to the report. These reports show up in the run report window along with other reports. However, on the calibration screen, selecting Print will bring up a list of only Run Time reports. In the Run Time, since you just completed a calibration, the specific CTAG is known. Run Time reports typically are written to report off the stored procedure MCCALRESULT or MCCALRESLOC.

Location Reports Location reports are designed to print information about a specific location event. You can run these reports at any time from the Manual Entry application. These reports are written around a specific LTAG value. In the Manual Entry application, the system will prompt the user for the asset number, then for a location event. From this information, a specific LTAG value can be passed to the report. These reports show up in the Run Report dialog, along with other reports. However, on the location screen, selecting Print will bring up a list of only location reports. Location reports are typically written to report off the stored procedure MCLTAG or MCLTAGINVENTORY.

Maintenance Reports Maintenance reports are designed to print information about a specific maintenance event. You can run these reports at any time from the Manual Entry application. These reports are written around a specific RTAG value. In the Manual Entry tool, the system will automatically prompt the user for the asset number, and then for a Maintenance event. From this information a specific RTAG value can be passed to the report. These reports show up in the “run report” window along with other reports. However, on the location screen, selecting print will bring up a list of only Maintenance reports. Location reports are typically written to report off of the stored procedure MCRTAG or MCRTAGINVENTORY

Inventory Reports Inventory reports are designed to print information about a specific Inventory item. You can run these reports at any time from the Manual Entry application. These reports are written around a specific MTAG value. In the Manual Entry tool, the system will automatically prompt the user for the asset number. From this information a specific MTAG value can be passed to the report. These reports show up in the "run report" window along with other reports. However, on the Inventory screen, selecting print will bring up a list of only Inventory reports. Inventory reports are typically written to report off of the stored procedure MCMTAG or MCMTAGINVENTORY.

Customer Reports Customer reports are designed to print information about a specific Customer. You can run these reports at any time from the Manual Entry application. These reports are written around a specific KTAG value. In the Manual Entry tool, the system will automatically prompt the user for customer ID. From this information a specific KTAG value can be passed to the report. These reports show up in the "run report" window along with other reports. However, on the Customer screen, selecting print will bring up a list of only customer reports. Customer reports are typically written to report off of the stored procedure MCKTAG or MCKTAGINVENTORY

All report types use the standard RPT extension.

Elements of a Report

Every report must specify the following three things:

1. The *source location* of the desired information. Specify the location by naming the data source, the table of interest, and the field name. In the Fluke Metrology system, field names are actually their field numbers, with a single alphabetic prefix identifying the table. For example, the Inventory table's stores the Asset Number in field 4201, so this field is identified by I4201.
2. *Selection criteria*. *Crystal Reports* lets you specify selection criteria using the Select Records Expert. It is not necessary to know SQL (Structured Query Language), but it is possible to edit SQL commands directly for record selection in complex situations. If you use *Crystal*, you have all the power of that language for specifying the precise information you want included in the report. If you are reporting off a stored procedure, no selection criteria is necessary as it is embedded in the stored procedure.

3. The *appearance* of the report. Again, *Crystal* has a friendly interface that lets you produce practically any appearance, from the very formal look of a Certificate of Calibration, complete with corporate logo and detailed laboratory specifications, through informal lists for your own use.

Running Reports

Once you have defined a report, you can preview and print it from within *Crystal*. You can also include the new report in the system and start running the report from the Manual Entry Application.

The Fluke Metrology software provides one additional capability you cannot get from *Crystal Reports* directly: prompting the user for values to control the report. Any General report can ask for up to five values to be substituted into a report's formulas. Each prompt can have a comment line associated with it.

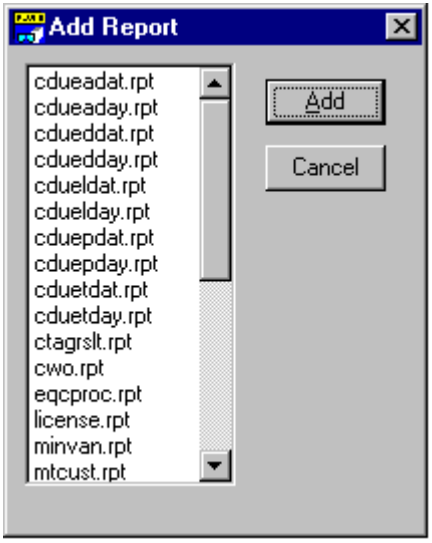
This capability, called auto-substitution, is also available for Run Time reports, but without the prompting. In this case, the substituted values are taken from the DBCONFIG table.

The *Crystal Reports* Print Engine uses values supplied by the operator for the printout. You can also save the dialog for use again later. It is important to note that the prompting mechanism relies on *named formulas* having a default value that is stored within the report definition itself.

Adding a Crystal Report to the System

The Fluke Metrology software provides a way to run reports directly from the Manual Entry application. You must use reports created with *Crystal Reports* Professional. You must copy the report definition file to the appropriate directory and add the report information into the report table in the databases. To accomplish this, proceed as follows:

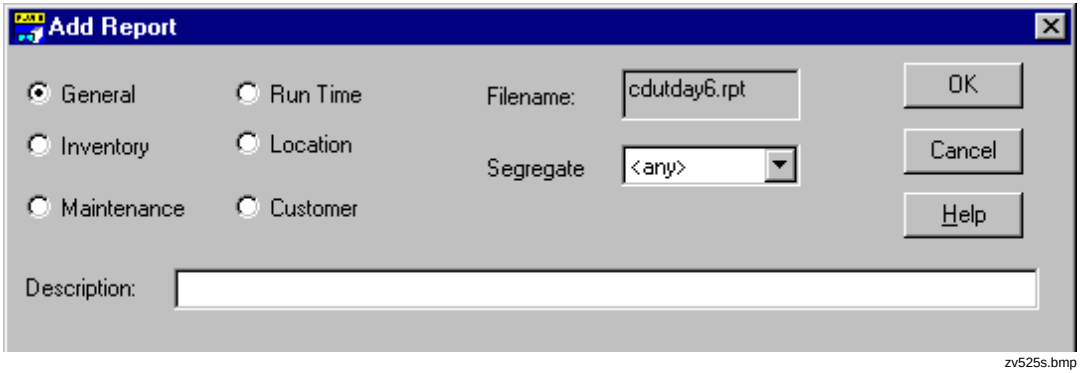
1. Copy the file to your main MET/BASE directory (typically C:\METCAL), and start the Manual Entry application.
2. Select **Reports**, **Add Report**. A report selection screen appears, as shown in the following figure.



zv524s.tif

Figure 5-1. Adding a Report in the Manual Entry Application

3. Use the pull down menu to select a file name to add.
4. All available **.RPT** files in the will appear for selection. Highlight one and press add. The following screen appears.



zv525s.bmp

Report Creation Strategies

To create your own customized reports, you can use a number of different approaches. *Crystal Reports* is a very capable report writing program, and it is the program that was used for developing the reports supplied with the Fluke Metrology software.

Using and Modifying Sample Reports

- The simplest strategy for generating a report is to use one of the supplied reports without modification. For the most part, these samples will cover the needs you most frequently encounter.
- The next way to approach the process is to modify an existing report. Starting with a fully-developed report can significantly reduce the time you need to spend to get the reports you need. With only minor modifications, you can quickly personalize one of the supplied sample reports.

  Even if you eventually decide to create a report that is not based on a supplied report, you should spend a little time in modifying one of the supplied reports to familiarize yourself with Crystal Reports, and to investigate the features of the sample reports. This is especially critical if you have little prior experience with report writing.

Creating Reports from Scratch

- The next developmental level is to create a report from scratch, making use of one of the "stored procedures." In this method, you use pre-defined database requests that are stored in the database itself. As part of the Metrology Software package, Fluke provides stored procedures that you can use to quickly develop custom reports. A list of these stored procedures and information about using them appears later in this chapter.
- Finally, you can create an entirely new report by accessing the database tables directly. Creating a report from the base tables requires you to be knowledgeable about the underlying database structure and may require further study. This may be the best option if you have complex reporting requirements, as long as you are aware that this approach requires more fundamental understanding of the database and may require more effort, depending on your experience level.

Using Other Applications for Writing Reports

Besides using *Crystal Reports* or some other report writing program, you can create your own reports using a spreadsheet or word processing application that can access the SQL database using ODBC. You could leverage your familiarity with Microsoft *Word 6.0* to create print-quality reports that meet your needs.

By using a query to filter, sort, and select specific fields, you can get exactly the information you want from the data source. To keep the data in your document up to date, you can create a link to the data source. Whenever the data is changed in the source file, Word 6.0 can automatically update it in your document.

1. To create a report in *Word 6.0*, start a new document with your preferred template. This makes available your own familiar macros and styles.
2. Place the cursor where you want to insert the data.

3. Under the **Insert** menu, go all the way to the last selection, and click **Database.**
4. Click **Get Data**.

If you want to use *Word 6.0* features to retrieve, sort and filter the data, select a data source, click **Open** and do the following:

1. Click **Query Options**.
2. Select the criteria you want and click **OK**.

If you want to use Microsoft *Query* to retrieve, sort, and filter data do the following:

1. Click MS Query
2. Construct your query.
3. To customize the Word 6.0 table formatting, click Table AutoFormat.
4. Select the options you want and click OK.
5. Click Insert Data.
6. Under Insert Records, select the records to include in the Word table.

If you want to be able to update the data in the Word 6.0 table when the source data changes, select Insert Data as field check box.

Word inserts the data in the table as a DATABASE field, which is linked to the source data. To update the field, click in the table and press F9.

Other Sources of Information

One critical element in your report creation strategy ought to be to familiarize yourself with the available sample reports and the report creation process. Besides this manual, other sources for more information include the *Crystal Reports* and *Sybase* manuals, both printed and online.

Before you spend too much time creating your first report, you should take a few moments to become familiar with the process by reading the remainder of this chapter, and then look through the other documentation to gain an idea of the process and capabilities available.

What is in the Rest of This Chapter

The remainder of this chapter discusses how to modify one of the supplied reports. It ends with some background information you will need if you must create a report entirely from scratch.

Section 2: Modifying Sample Reports

The reports that most users create are based on the reports that have been provided as part of the base package. Often, you can use these reports just as they come or with only slight changes. In either case, the supplied reports can serve as valuable templates and examples. Creating a new report is fairly simple if you base it on one of the provided reports.

After you identify the changes you wish to make to a supplied report, open a copy of it in *Crystal* to modify it as needed. *Crystal Reports* provides a Design mode and a Preview mode. This allows you to check the affects of any changes without needing to run the report from one of the Fluke Metrology applications.

The remainder of this section is a tutorial exercise that guides you through the steps for modifying one of the reports that has been supplied with the Fluke Metrology software.

The tutorial exercise addresses the major issues faced by users when they want to modify one of the supplied reports:

- Selecting an existing report and opening it in *Crystal*'s Design Mode.
- Adding a graphic such as a corporate logo or emblem.
- Adding a text object to print out boilerplate material, such as a legal declaration.
- Adding page numbers for multi-page reports.
- Generating a total page count so that each page carries an inscription, as in "Page ***x*** of ***y***."

Selecting a Report to Modify

To begin to modify an existing report, proceed as follows:

1. Make sure that the database engine is running. Then, start up *Crystal Reports*. The first time you start up *Crystal*, it displays an opening screen called the Welcome Dialog. This screen provides you the option to create a new report or to open an existing one.

   At the bottom of the Welcome Dialog, a selection lets you decide whether to continue to start *Crystal* with this dialog or go immediately to the menus. If you turn this option OFF, each time you start Crystal subsequently, you must use the File menu to select either **New** or **Open**.

   "New" means that you will be creating a report from the ground up, while "Open" means that you want to start by modifying an existing report.

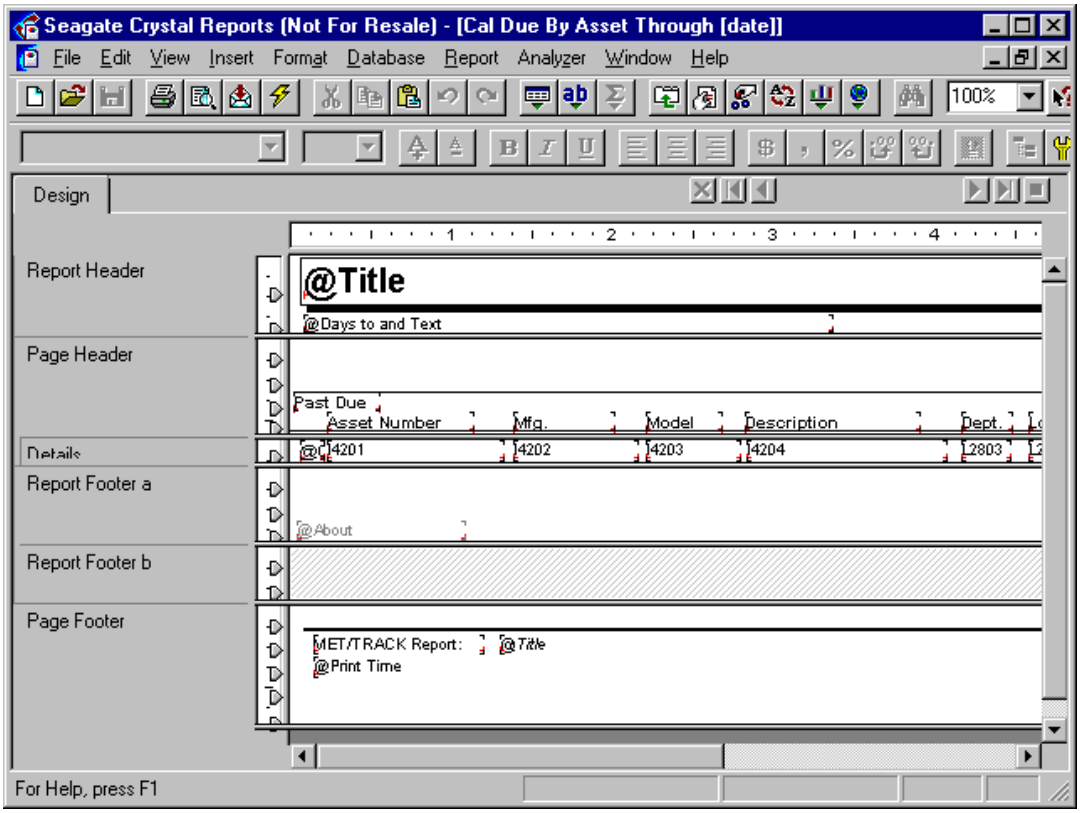
2. Whether you leave the Welcome Dialog active or not, you must first choose between creating a new report or opening an existing one. If you select **Open**

Report, *Crystal* presents a standard directory and file-selection dialog for you to browse to the RPT file you wish to work on.

If you have de-activated the Welcome Dialog, go to the File menu and select **Open**.

3. Browse to C:\METCAL\CDUEADAT.RPT. (This is the default path and filename. Adjust as needed to conform to your local installation.)

   The following illustration shows the calibration due report **CDUEADAT.RPT**, as it appears when opened in *Crystal Reports* Design mode.



zv501s.tif

Figure 5-2. Calibration Due Report Opened in Crystal Reports Design Mode

Make note of the following:

- This display may differ from that on your computer. Instead of the field names, you may see lines of xxxxx's. Proceed as follows: open the File menu, and select **Options**. A dialog box appears. In the lower right hand corner, under "Field Options", add a check mark to the box labeled "Show Field Names."

- The supplied reports use strict naming conventions. You may see some reports with a letter "J" (RTRSLTJ1.RPT for example). These are Japanese versions. Open these reports only if you have a Japanese system.

As you view the display, do not be too concerned just yet with the appearance of the viewing area (*Crystal* calls this the "Design Window"). The next section describes the process of creating a completely new report. There, you will find an illustration of the Design Window that identifies all the parts and describes their use. In this first tutorial, the idea is to leave the basic parts as they are, and just get practice with changing some of the details.

Both selections can be useful, because, at times you may want to work only with the layout of the report. By de-activating Field Names, you can avoid being distracted by all the text. At other times, you must work with the individual fields, so you want to see the field names.

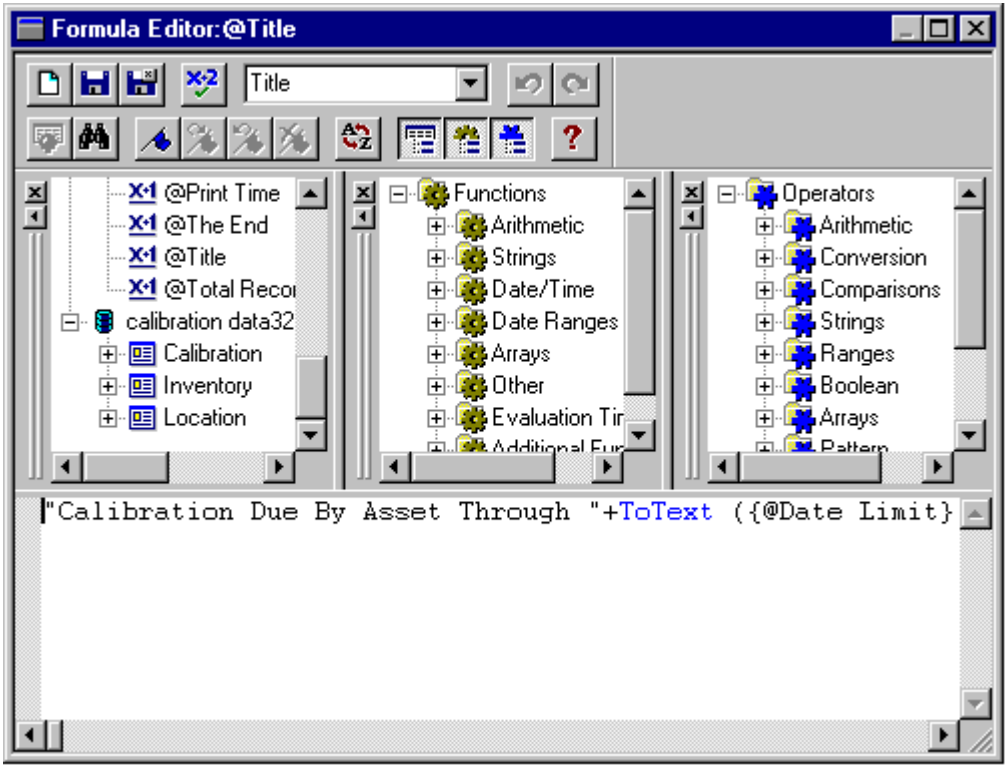
Using a Formula to Create Conditional Headers

In the CDUEADAT.RPT example, the report heading is actually generated by a report formula, so you must edit the formula to change the text. In this case, the Report Header is being generated by a formula named **@Title**. All of *Crystal*'s named formulas are identified by starting with the @ character.

However, headers can be placed in text objects, too, so depending on how the report is set up, you may need to edit a text object rather than a formula. The following describes how you would proceed in either case.

1. Click once in the upper portion of the report, near **@Title.** This selects the formula.
2. Now click the right mouse button once. A floating menu appears. The menu contains selections appropriate to the currently selected object. Select **Edit Formula .**

The display should look similar to the following figure:



zv502s.tif

Figure 5-3. Editing the Formula that Creates a Due-For-Calibration Report

In this case, the report header is being generated by the formula:

```
"Calibration Due By Asset Through" + ToText({@Date Limit})
```

To understand how this formula is constructed, you need to know the following about how *Crystal* formulas are constructed:

- Text in quotes, like `"Calibration Due By Asset Through"` is just text. The characters inside the quote marks will appear exactly in the header.
- The Plus Sign (+) means to add the foregoing text to whatever comes next in the formula.
- The conversion function "To Text" converts fractional numbers, currency, and date/time values to text strings.
- The syntax ({@Date Limit}) means that the result of *another* formula called @Date Limit is to be replaced into the header. The "To Text" function converts the numbers into text for replacement into the report header that is being built.

- The @ Date Limit formula is nested within the @Title formula. To edit a nested formula, you must begin to insert it. Proceed as follows:

  1. Exit the Formula Editor by clicking **Accept** or **Cancel**.
  2. Under the **Insert** menu, select **Formula Field**.
  3. From the list, select **Date Limit**, then **Edit**.
  4. Edit the formula, then dismiss the **Insert** dialog. This is the only way you can change a formula within a formula.

     In the case of @ Date Limit, the meaningful item to edit is the date. In this case, the date you place into the formula is the ending date -- the date through which you wish to see calibration due dates.
  5. Finally, when you add this report into MET/TRACK, you can set up a prompt for "Date Limit" that will substitute the operator's input for the value in the formula.

The formula in this example translates into, "Put the words 'Calibration Due By Asset Through', into the Report Header. Next, convert to a text string and add in the result of the embedded @Date Limit formula."

The foregoing exercise shows how to modify a report title based on a date. In Step 4 above, if you changed the date to read 7/1/97, the resulting title would read as follows:

Calibration Due By Asset Through 7/1/97

If @Date Limit is prompted for, whatever value the operator types in will be substituted into the Header and anywhere else the @Date Limit formula is called for in the report.

Headers in Text Objects

If the header is in a *text object* (rather than in a formula), you would select **Edit Text Object** instead of **Edit Formula.**

In this case, a ruler line would appear above the selected text. You can add text, move or modify the ruler lines, and format the text by clicking the right mouse button, or by using the toolbar at the bottom of the window.

For example, the heading, "Past Due" in the Page Header area is contained within a text object. If you select that text, then click the right mouse button, a floating menu appears. Selecting **Edit Text Object** (the first choice on the menu) lets you change the text in the text object, or modify any of its attributes — size, font, style (bold, italic), and so on.

In the example report, the Past Due heading is in 8 point Helvetica Bold.

Adding a Graphic

Personalizing a report for your organization or for a particular customer can result in a very attractive presentation by the simple addition of a graphic, such as a logo.

You can add graphics to reports in several different formats, including bitmaps, (BMP), graphics interchange format (GIF), PC Paintbrush (PCX), tagged image format (TIF), and Targa (TGA). These five formats cover the most commonly used graphics formats, and many applications can be used for generating them.

To add a graphic to a report, proceed as follows:

1. First, be sure that the database engine is running. Then, start up *Crystal Reports*.
2. If the Welcome Dialog is active, choose **Open Report**. If it is not, under the **File** menu, select **Open**.
3. Browse to the location of the report you wish to change.
4. When the report opens, go to the **Insert** menu and select **Picture**.

   A dialog box opens for you to select or browse to the location of the graphics file.
5. Select the desired graphics file. When you click **OK** to dismiss the dialog box, a rectangle appears in the report that follows the movement of the mouse.
6. Position the rectangle where you want the graphic to appear, and click the mouse button once. The image from the file appears at that location in the report.

   You can crop, size or scale the image, or reposition it by dragging it to a different location. If you need more powerful graphics capabilities (such as rotating an image, or changing some of its colors), perform those operations in your graphics editor before bringing the illustration into your report.
7. To delete an image, select it and press the **DEL**ete or backspace key. If you do this, note that *Crystal Reports* has added space to accommodate the image, so you may need to remove that space if you remove the image. Similarly, if you insert a very large graphic and size it to be smaller, you may have to delete the added space.

   To do that, position the cursor in the middle of the line separating the report elements. When the cursor changes to up-and-down arrows, move the horizontal bar to re-size the element.

As is true of any element you add to a report, if an image is added to the “TITLE” area, it only prints on the first page of any multiple-page reports. If you want the image to repeat on every page, you must include it in the page header or page footer area.

Adding Boilerplate Text

Text objects provide a convenient way to add arbitrary text to the reports you create. A common use of a text object is to insert the same text repeatedly. This is called *boilerplate* text.

For example, you can add a text object to every page of a report to describe what type of report it is, or to make some sort of legal declaration, such as a claim regarding liability limits, or to explain the traceability of instruments in your lab.

To add a text object to a report, proceed as follows:

1. First make sure the database engine is running, then start up *Crystal Reports*. Under the **File** menu, select **Open**, and browse to the location of the report you wish to change.
2. When the report opens, go to the **Insert** menu and select **Text Object**.

   A rectangle appears in the report that follows the movement of the mouse.
3. Move the rectangle to the area where you want the text to appear: *Title area* for first page only, *Page Header* for the top of every page, and so on. When the rectangle is in the desired location, click the left mouse button once. This sets the location of the text object.

   When you establish the location of the text object, the cursor moves into the selected text area, and the text object editor starts up. A ruler line should display above the selected text area.
4. In the text box, type in the text that you want to appear in the text object. When you have finished typing in the text, let the text object stay selected.
5. Now, click once on the right mouse button. A floating menu appears.

   From this menu, you can elect to change the formatting in many different ways. Most of the available selections are self-explanatory and lead to familiar dialog boxes.

   For example, if you select **Change Font**, you can change the font type, size, and various attributes such as bolding, italic, and the displayed color. Explore the dialog box selections. You have a high degree of control over the appearance of text that appears in the report.

In addition, when you are editing a Text Object, you ought to explore the options available under the Format menu. The *Format Text* selection provides a large variety of options for various needs. For example, under the “Common” tab, the *Can Grow* selection lets text objects grow to fit the size required by the incoming information. *Crystal Reports* provides many such features. You can even do conditional formatting. For example, you can print the results of failed tests in red and passed tests in blue.

Adding a Page Number

You can easily add page numbering to multi-page reports. The following steps result in a simple serial page number added to each page in the report.

1. First, make sure the database engine is running. Then, start up *Crystal Reports*. If the Welcome Dialog is active, select Open Report. If you have de-activated the Welcome Dialog, go to the **File** menu and select **Open**. Browse to the location of the report you wish to change.
2. From the Insert menu, select **Special Field**, Then, continue dragging across to the submenu and select **Page Number Field**.

   When you click **OK** to dismiss the dialog box, a rectangle appears in the report that follows the movement of the mouse.
3. Move the rectangle to the area where you want the page number to appear (normally the Page Footer area).
4. When the rectangle is in the desired location, click the left mouse button once. The rectangle changes into a box containing the words "Page Number." If you have disabled the Show Field Names option, you will see a representative number, such as 5,555,555.
5. With the Page Number Field selected, click once on the right mouse button.

   A floating menu appears with formatting options similar to those available for text objects.

   - To format page numbers, proceed just as you do for a text object: highlight the field and click the right mouse button to access the floating menu that makes the formatting options available.
   - You can also use the toolbar at the bottom of the window to change fonts, paragraph alignment, and so on.

Generating a Total Page Count

Any report you create using *Crystal Reports* can generate a total count, for those situations when your report must indicate, for example, "Page **x** of **y**" on every page.

To insert a Total Page Count field, proceed as follows:

1. First. make sure the database engine is running. Then, start up *Crystal Reports*. Under the **File** menu, select **Open**, and browse to the location of the report you wish to change.
2. From the **Insert** menu select **Special Field.** Then, continue dragging across to access the submenu. From the submenu, select **Page Number Field**.

3. A rectangle appears that follows the mouse movements. Position the Page Number Field (normally in the Page Footer area). Then, click the left mouse button once.
4. With the new field selected, click the right mouse button to access the formatting options. Page numbers should be integers, so make sure that the *Decimals* selection is set to "1."
5. Insert two text objects, the first containing the single word "Page", and the second containing the single word "of." Position the fields immediately before and after the Page Number Field you just inserted.
6. From the Insert menu, again select **Special Field**. Drag across to the right to select **Total Page Count**.

Note

The total page count capability may not display the correct page count if you use features that cause page breaks and that require renumbering pages after the page breaks. If you need both the total page count and page breaks, explore the Crystal Reports *online help for guidance.*

Considerations Specific to Run Time Reports

The following topics apply only to Run Time reports.

When run immediately following a Calibration Procedure, Run Time reports return information about one and only one calibration event, the one just completed. When run from the Manual Entry application, a dialog box asks the user for an asset number and then for a calibration event. After you enter the asset number, the entire list of calibration events for that asset appears. The list identifies date and time of the calibration, the Pass/Fail status, and the name of the Calibration Procedure used. Except for this once case in the Manual Entry application, Run Time Reports do not support operator prompts.

Configuring PostRun Reports

Up to eight Run Time reports can be set up for selection by the operator to run after a calibration. PostRun reports are configured in the Run Time application. Proceed as follows:

1. With the database engine running, start the Run Time application.
2. When the Run Time application starts, pull down the **Configure** menu, and select **PostRun Reports.**

   The Run Time presents a dialog box listing all available reports on the left. The report you wish to add must exist in the METCAL directory and must already be added using the Manual Entry application.
3. Select the report from the list on the left, and press the **–>** button to add it to the list of reports available. All the reports that appear in the list on the right will be available for operator selection at the conclusion of the calibration.

Run Time Report Rules

- Must be based on **ctag** as the only record selection criteria. (See Section 5 later in this chapter for a report that can list all valid ctags in your database.)
- Can read database tables directly, or use a stored procedure.
- Formula @mt *company* substitutes the company field from the **Configure Company** dialog box in DB Setup.
- Formulas *@mt user1* through cause the Run Time application to substitute user1 through user5 from the **Configure | Company** dialog box in DB Setup.

Section 3: Creating Reports from Scratch

This final section of the chapter about reports provides important background material about the overall reporting process. This information is most useful if you are creating reports entirely from scratch, although it may be of interest even if you are simply modifying supplied reports.

If you are creating new reports (not based on one of those supplied with the Fluke Metrology Software package), be sure to refer to the tutorial information provided in the Application Notes on the CD. Look in the DOC directory for the files called NEWRTRPT.PDF and NEWGNRPT.PDF for guidance in creating a new Run Time report or a new General report.

Report Creation Process Overview

Creating your own reports can be very simple, or it can be quite complex, depending on your needs and the effort you are able to expend. The following are the major steps you must take, no matter how you approach the process:

- **Design the report.**

  The first step is to design the report. What information is it to contain? Is it to be run only one time, or according to some schedule? Is it a general report (one that can run any time), or a Run Time report (one that automatically runs at the end of a calibration procedure)?

  Before considering the method you will employ to create the report, first have a purpose firmly in mind, asking yourself, "What exactly is this report supposed to accomplish?" By designing the report first, you begin to identify which database fields contain the information of interest. You also make a start in deciding on a plan of attack: whether to duplicate the report from an existing one, or to build it entirely from scratch.

- **Create the report.**

  Once you know what you want to do, decide upon a plan of attack for creating the report. More than likely, you can modify one of those supplied with the Fluke Metrology software.

  In *Crystal Reports*, this is very straightforward. Open the sample report in *Crystal*, and make an necessary changes as described in the prior section of this chapter.

  If you are creating a new report, first open *Crystal*, then select **File Options**. Open the **SQL** tab, and toggle the selection ON that allows reporting on stored procedures. Now, as you begin creating the report, the stored procedures are listed and available to the report, just as the database tables are.

- **Add the report to the system.**

  After saving the report from *Crystal*, the final part of making the report is to add it to the system. Note that ONLY *Crystal Reports* can be added. If you are using some other report package, you will not be able to run the reports from within any of the applications.

  Reports are added to the system using the Manual Entry application. Use the Run Time application to configure Run Time reports for Post Run use if you want it to be available to operators upon completion of a Calibration Procedure.

How Information Gets Into the Database

The database gets its information from three sources:

Manual Entry	Information can be explicitly entered into the database by manually keying it in or wanding it in with a bar code reader.
Importation	Information that pre-exists in a data source can be imported into the database.
Procedure Results	Data goes into the database automatically any time you run a calibration procedure using the Run Time application. For example, the date and time might be retrieved from the current system date and time. This method is of interest only if you are using MET/CAL or 5500/CAL for automating your calibrations.

Database Table Structures

Fluke Metrology software uses a relational database consisting of many tables interrelated by key fields. The software uses some of the tables internally, and you should rarely, if ever, need to access them. The primary tables for producing reports on assets are as follows:

mt.customers	Information about your customers.
mt.inventory	Information about your assets.
mt.calibration	Latest and historical information about all Calibration events.
mt.calresults	Step by step test results for Calibration events.
mt.standards	All of the standards used to perform calibrations.
mt.location	The place where the asset is currently located
mt.repair	Historical information about an asset’s Maintenance events.

How to Interpret the Table Structures Appendix

Appendix C of this manual, "Table Definitions for Reports," illustrates the database tables and their relationships to each other.

Note

The tables in Appendix C reflect the database as it exists when the product is shipped. If you perform any customization, this information will not be correct for your installation. These tables were generated using a supplied report, TBLDEFF.RPT (Table Definitions Report). After customization, it might be appropriate for you to run TBLDEFF.RPT to generate a corrected version of the information in this appendix.

The appendix does not describe all the tables in the database. The database engine uses system tables, and the software itself creates and maintains a number of tables for its own internal uses. Even of the tables shown, only some fields will be of interest to you. However, for effective report writing, you need some basic information about the position of fields, their contents, and some customization information.

For example, assume that you wanted to produce a report that shows the current system customization. By looking at the table structure, you can see the Customization table contains the kind of information that such a report might contain. By knowing such things as the maximum possible length the field, you can format your report to present the data clearly.

The Customization table contains information about field-level customization. The column names represent a particular field attribute (not a number.) Attributes define each field; the Customization table stores the present settings. Actually, the database uses some preset field attributes (e.g., FLDNUM for field number and DATALEN for data length). You can only make changes to the attributes via the customization interface.

The center column of each table has a short description of the intended use for the field. The description may be an actual field title or a few words to relay the idea behind including the field on the form. The last column gives the data type and byte size of the field. The data types are Character, Date, DOUBle, INTeger, and TimeStamp. Date fields contain only month, day, and year, while the TS field adds hour, minute, and second.

The arrows show the internal linkages within the relational database. Two fields (CTAG and MTAG) carry linking information. The links are not customizable, and you cannot affect their action externally. The software maintains the links automatically.

Stored Procedures

A number of stored procedures are provided as a part of the Fluke Metrology Software package. You can find the input and output parameters of these stored procedures by running the report PROCPARM.RPT. This will generate the tables as they exist when the product is shipped. After customization, you might want to run PROCPSRM.RPT to generate a corrected list of tables.

Not all of the reports supplied with Fluke Metrology software access the base tables directly. Instead, a technique has been developed to run a "stored procedure." Stored procedures run within the database. They scan the base tables for information, and return this information to the report writer program for formatting and printing.

A stored procedure is a series of SQL commands stored in the database. There are several reasons for storing such a series of commands in the database. Following are the most important benefits from using this approach:

- First, the stored procedure runs in the database itself, so network traffic is significantly reduced. After it executes, the stored procedure returns only the requested data to the client. This makes the stored procedure run much faster than the equivalent request coming from a report searching the base tables. It also uses the network bandwidth much more efficiently.
- Secondly, the stored procedure pre-processes the information, so the amount of data returned is only a subset of the entire body of information contained in the Calibration database. As a consequence, the sorting and grouping done by the report is minimized, resulting in speedier, more efficient operation.
- Finally, *Crystal Reports* does not perform multiple SELECT statements, so if you need multiple records from more than one tables in the database, the best solution is to use a stored procedure.

Fluke provides the following stored procedures with the software.

mc_cal_due	Provides a list of instruments due for calibration.
mc_cal_loc_due	Provides a list of instruments due for calibration with location included.
mccalresult	Gets inventory, cal summary, stds, and results for a CTAG.
mccalresloc	Gets inv, cal summary, stds, results and loc for a CTAG.
mcltag	gets location info for an ltag.
mcmtag	gets inventory for an mtag.
mcktag	gets customer info for a ktag.
mcrtag	gets maintenance info for an rtag.
mcltaginventory	gets location and inventory info for an ltag.
mcrtaginventory	gets maintenance and inventory info for an rtag.
mcmtagcustomer	gets inventory and customer for an mtag.
mcktaginventory	gets inventory and customer info for a ktag. (multiple rows)

How to Interpret the Results Structures Appendix

Appendix F of this manual illustrates the structures of the results information returned by the stored procedures. You will need this information if you wish to use a stored procedure to filter and supply needed information. The tables in Appendix F describe the fields whose values are returned by the stored procedures.

Note

The tables in Appendix F reflect the database as it exists when the product is shipped. If you perform any customization, this information will not be correct for your installation. These tables were generated using a supplied report, PROCPARM.RPT (Procedure Parameters Report). After customization, it might be appropriate for you to run PROCPARM.RPT to generate a corrected version of the information in this appendix.

Run Time reports can be set up to run automatically upon completion of a Calibration Procedure. Those reports that are based on the MCCTAGResult stored procedure automatically supply the *ctag* value, since the Run Time has just assigned it.

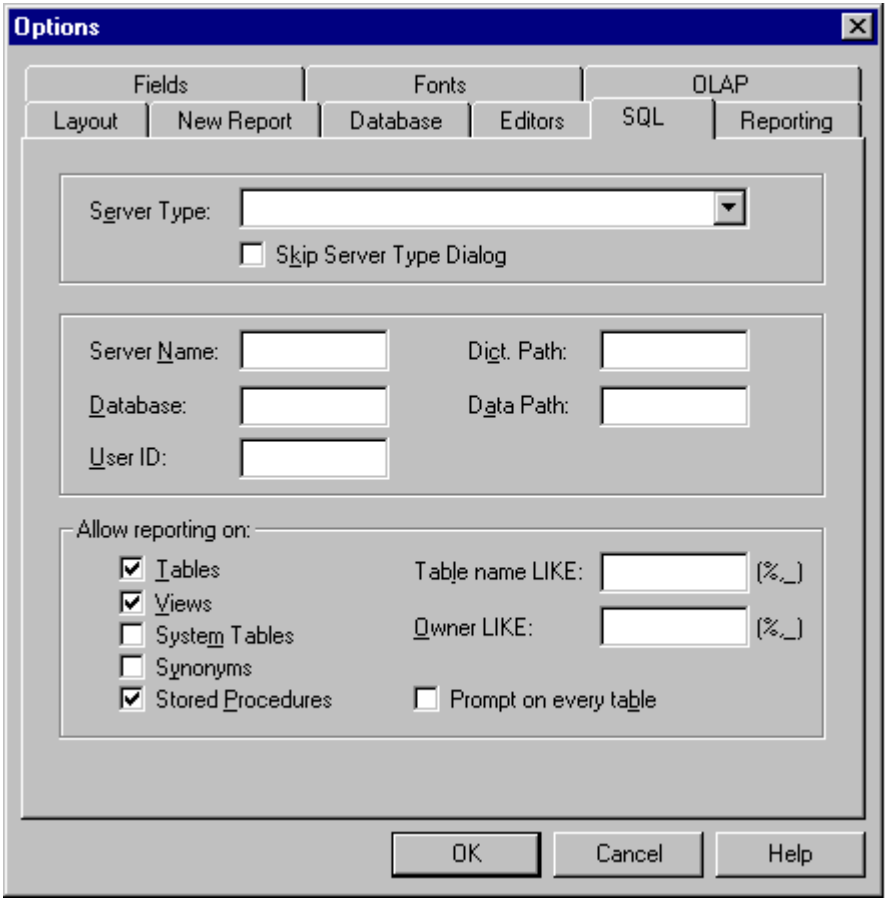
When a report is designed for use only as a Run Time report, it takes one argument (the *ctag* that the Calibration Procedure has just assigned). If you wish to run a Run Time report later (as if it were a General report), you must supply the *ctag*. This is the only time a Run Time report prompts for information, and prompting for the *ctag* is an automatic function in this case. Normally, *ctag* values are not very visible – to find the *ctag* of a particular calibration event requires running another report specifically to find out valid *ctag* values. A report to accomplish this is provided, and is explained in the final section of this chapter.

Another Fluke-supplied stored procedure, MCResult, takes three arguments: the asset number of interest, and a beginning and ending date. This procedure is intended to be used for General Reports, because only through the report prompting mechanism can the dates be supplied to the procedure.

Crystal Setup

The following directions set a number of options in Crystal Reports that some report developers have found to be helpful. None of the settings discussed here make any permanent alteration of the system, so, if you find you prefer some other behavior, it is an easy enough to undo the change.

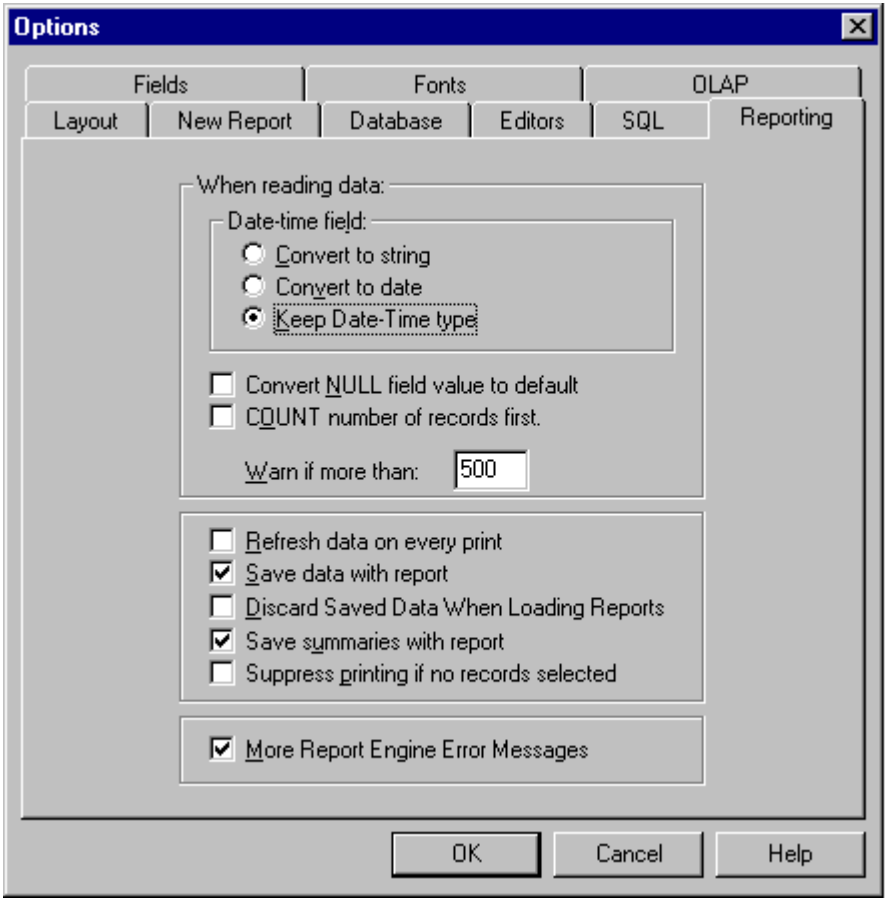
1. After you start up *Crystal*, select **File | Options**. Open the **SQL** tab and select the option to create new reports from SQL Tables. The tables in the Metrology database are SQL tables, so when you create a new report, the tables will be available as soon as you specify the database. You save a few steps by not having to make this selection every time.



zv522s.tif

Figure 5-4. Crystal Reports File Options - SQL

2. Open the **Reporting** tab, and make sure the option **Convert NULL field value to default** is de-selected. If selected, this option prevents searching for NULL values in the database, which can make generating some reports more difficult.



zv523s.tif

Figure 5-5. Crystal Reports File Options Reporting Tab

3. While the File Options dialog box is opened, another selection can simplify your later work with Crystal.
4. Under the **Layout** Tab, place a select mark in the option "Show Field Names."

   With this selection checked, the Design Window shows the names of the fields and formulas at the location where they are places, rather than just indicating their location by a string of xxxxxx's.
5. Finally, after you close the File Options dialog box, make one final setup step. Under the **Format** menu, select **Section** (the last selection in the menu). You will be presented with a list of all the sections in a report. Select **Details**, and, when the formatting options menu displays, select the option to Suppress Blank Lines.
6. The effect of making this selection is that any reports you generate will already have blank line suppression enabled.

Automated Report Control Prompts

Crystal Reports provides for the use of ***Formula Fields*** (fields that you can print and use to determine the final value of other formula fields or to modify the record selection formula.) You name Formula Fields when you define them. Up to five of these named formula fields can be identified when you add a report to the system, together with an appropriate prompt (for example, "Enter the asset number of interest") and a comment. The Run Time application can then use the prompt and comment to open a dialog with the user. It then passes the user-entered value to the report engine to replace the value assigned during report creation for this particular print job. If the user-entered value needs to be a date, type in the value in the following format: DATE(yyyy,mm,dd). For example, to substitute June 1st, 1997, you would type the following:

```
DATE(1997,06,01)
```

A General Report to Find Valid CTAGs

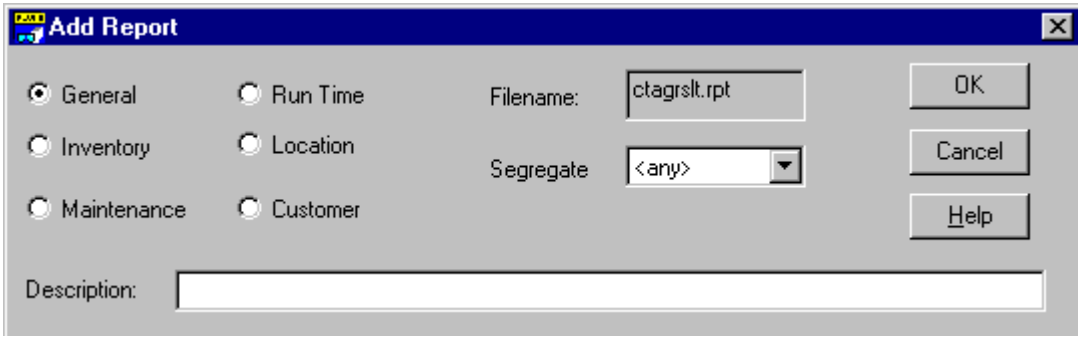
The final topic in this chapter presents a General Report that can be used for obtaining a list of all valid *ctag*s in the database. This is provided because one of the rules for a Run Time report is that it must be based on a single calibration event, as identified by a *ctag*.

Before creating a Run Time report, it is useful to have a list of *ctag*s in the database, since you must supply a *ctag* as a parameter for the MCCtagResult stored procedure. A General report (filename ctagrslt.rpt) has been supplied for the purposes of this exercise, but as shipped, the report has not yet been added to the system.

To add the report, proceed as follows:

1. Start the database engine, if it is not already running, and connect to the Calibration Database.
2. Start up the Manual Entry application. Under the Reports menu, select **Add Report**.
3. From the list in the upper left, select ctagrslt.rpt, the filename of the report that lists all the ctags in your database.

4. Click **Add** to dismiss the dialog box and add the report to the system.
5. Once the CTAG Results report is in the system, the Manual Entry application opens an Add Report dialog box that allows you to customize the report.
6. Customize the report using the Add Report Dialog Box.



zv510s.bmp

Figure 5-6. Add Report Dialog Box

Chapter 6
Procedure Editor

Contents **Page**

Introduction

The Procedure Editor application is designed for writing MET/CAL or 5500/CAL calibration procedures, but it also has a text mode, so you can use it for modifying any ASCII files, such as the control files that are discussed in Chapter 7.

The Editor lets you create new procedures or modify existing procedures. You can also compile, debug and test run calibration procedures.

In addition, the Editor allows you to configure the system instruments: add them to the system, assign their IEEE-488 addresses, and so on. The Tools menu provides a convenient way to start up other Fluke Metrology applications.

Logging Into the Editor

Note

You must log in at a level sufficient to use the Editor. The required level is configurable and is stored in the database. Refer to the DB Setup chapter for details.

To log into the Editor, first make sure that the database engine is running, then proceed as follows:

1. In the Fluke Metrology Software program group, double click the **Editor** icon. (Windows NT and Windows 95: Start Menu, Fluke Metrology Software.)
2. The editor login dialog box appears.

   In the User Name field, type in your user name, tab to the Password field, and type in your password. Click **OK**.

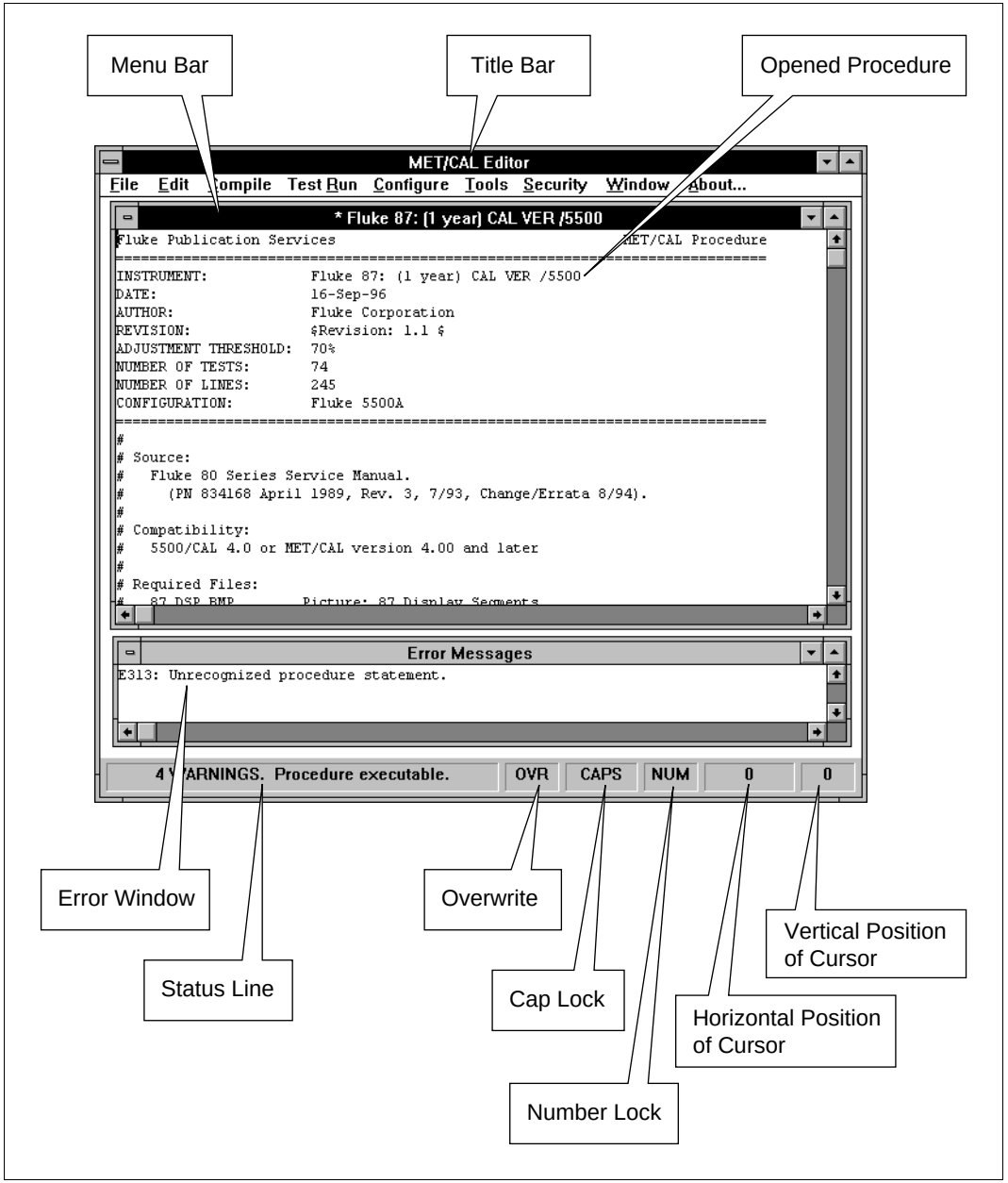
   If you do not have a password, see the administrator of the system.

You now have access to the Editor. Figure 6-1 is an example of the Editor window with a Calibration Procedure opened.

Refer to the following list for descriptions of the items labeled in Figure 6-1.

Title Bar	Shows the name of the application.
Menu Bar	Lists the available options in the Editor. Click the menu items to see the selections available.
Error Window	Shows you the error number and describes what the error is in the procedure you tried to compile.
Status Line	Describes what is taking place within the Editor. In Figure 6-1, the status line indicates that the 43 lines in the selected procedure have been loaded into the Editor.

Ovr	This annunciator indicates the current status of the overwrite/insert mode. When you press the Insert key, you can write over text in your procedure. If Ovr is not on, you can insert text into your procedure.
Cap	This annunciator indicates whether the Caps Lock function is on. When you press the Caps Lock key, you can type in all capital letters in the procedure.
Num	This annunciator indicates whether the Num Lock function is on. When you press the Num Lock key, you can enter numbers from the numeric keypad on the keyboard.
Horizontal	A position indicator that shows the cursor's horizontal position in the procedure. The numbers indicate the cursor position as the number of character positions from the left within the procedure.
Vertical	A position indicator that shows the cursor's vertical position in the procedure. The numbers indicate the cursor position as the number of lines from the top of the procedure.



il601c.eps

Figure 6-1. Editor Window with a Procedure Opened

Checking the User Level

After logging in, you can easily check your user access level.

To check your access level, click **About** in the Menu Bar. The **About** box show the current user name and user level.

If the user level is too low to allow access to Editor functions, see the administrator of the system.

Security

The Security menu provides a way to change your password.

To change your password, click **Security** in the menu bar, then **Password**.

The Change Password dialog box displays. Enter your old password, tab down to the New Password field and enter your new password.

Tab down to the Confirm New Password field and enter your new password again, then click **OK**.

Exiting the Editor

To exit the Editor, select **File**, then **Exit**.

If you have any open files that have been modified, you are asked if you want to save them before the Editor closes.

Synopsis of Editor Functions

The following table lists all the Editor commands in a quick-reference format.

Table 6-1. Overview of Editor Functions

Action	Keystrokes	Menu
Navigating		
Move left	←	
Move right	→	
Move up	↑	
Move down	↓	
Move to the beginning of the line	Home	
Move to the end of line	End	
Move to the beginning of the file	Ctrl Home	
Move to the end of file	Ctrl End	
Page up	Page Up	
Page down	Page Down	
Editing		
Cut selected text to clipboard	Ctrl X	Edit, Cut
Copy selected text to clipboard	Ctrl C	Edit, Copy
Copy to file	Ctrl W	Edit, Copy to
Paste text after cutting or copying from one source to another	Ctrl V	Edit, Paste.
Insert file	Ctrl G	Edit, Insert File
Delete selected text.	Delete	Edit, Delete
Searching		
Find strings in a file	Alt F3	Edit, Find
Find the next string	F3	Edit, Find Next
Replace strings in a file	Ctrl F3	Edit, Replace
Formatting a procedure file		
Create an adjustment step	Ctrl A	Edit, Adjust
Break a line at the cursor position	Ctrl ↵ Enter	Edit, Break
Continue a message on two or more lines	F6	Edit, Continue

Table 6-1. Overview of Editor Functions (cont.)

Testing a procedure		
Test run from current line	Alt F2	Test Run, Go
Test run from top of procedure	F2	Test Run, Restart
Make test run top window	Ctrl F2	Test Run, Show
File Handling		
Open a new procedure	Ctrl N	File, New
Open a previously written procedure	Ctrl O	File, Open Procedure
Save a procedure	Ctrl S	File, Save Procedure
Open a text file	Ctrl F	File, Open Text File
Save a text file	Ctrl J	File, Save Text File
Save a text file with a new file name		File, Save As
Close a file	Ctrl F4	File, Close
Print a file	Ctrl P	File, Print
Hide a procedure		File, Hide Procedures
Exit from the Editor	F4 I	File, Exit
Compiling a Procedure		
Compile the current procedure line	F8	Compile, Check Line
Compile an entire procedure forwards	F9	Compile, Next Error
Compile an entire procedure backwards	⇧Shift F9	Compile, Previous Error
Re-enable all warning messages	Ctrl F9	Compile, Re-warn
Configure		
Disable the compiler	Ctrl M	Configure, Text Mode
Select fonts		Configure, Font Configure, Default Font Configure, Display Font
Add and edit instrument configuration	F12	Configure, Instruments
Add and edit IEEE-488 boards		Configure, IEEE-488 Boards
Select a default Serial Port		Configure, Default Serial Port

Table 6-1. Overview of Editor Functions (cont.)

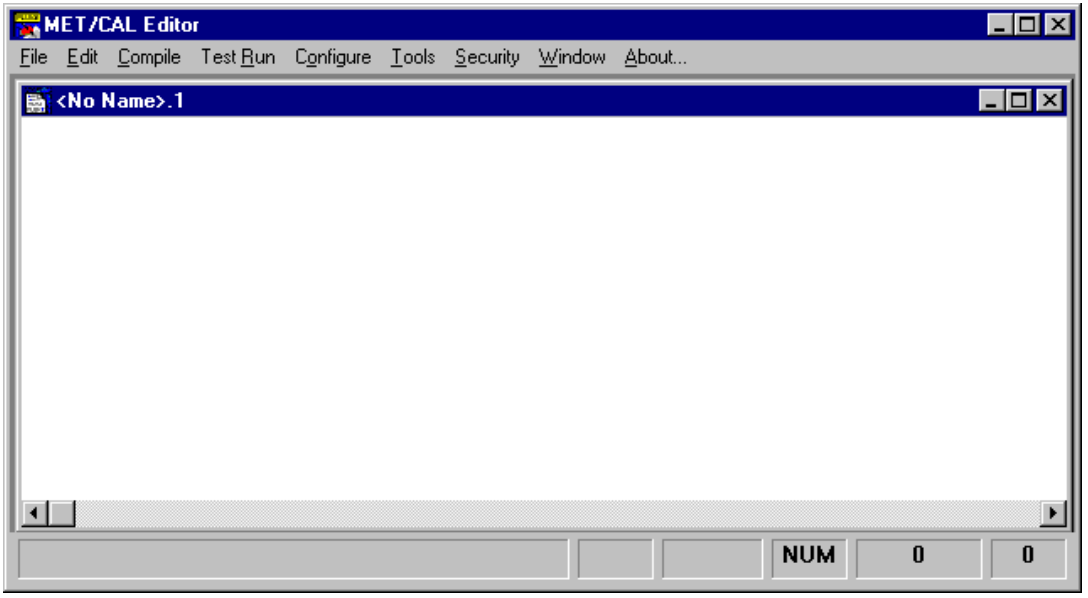
Tools		
Copy procedures Delete procedures		Tools, Copy Procedures Tools, Delete Procedures
Security		
Change your password without closing the application		Security, Password

Opening, Creating, and Saving Procedures

The File menu lets you open procedures and text files, save procedures and text files, print files, configure the print setup and layout, and exit.

Creating a New Procedure

New lets you create a new procedure. To create a new procedure, from the Editor menu bar, click **File**, then **New**. You should see a window titled <No Name>.1 at the top. Figure 6-2 shows what the New window looks like.



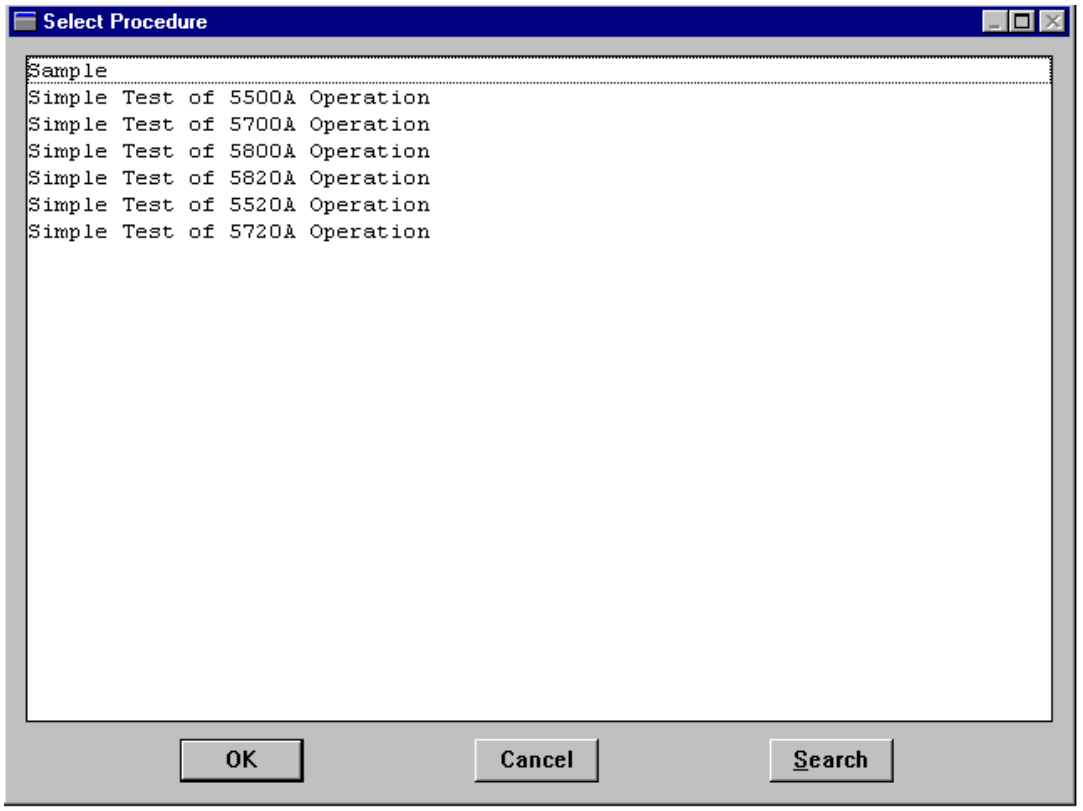
zv602s.tif

Figure 6-2. Opened New Procedure Window

Selecting a Procedure

Selecting **Open Procedure** on the menu opens a Select Procedure dialog box. This box lists the procedures that exist in the MET/CAL procedure directory.

To open a procedure, click **File**, then **Open Procedure**. Select the procedure you want. The following figure shows the Select Procedure dialog box.



zv603s.tif

Figure 6-3. Open Procedure Dialog Box

Click **OK**. A message tells you that the selected procedure is being loaded, then the procedure listing appears in the Editor window.

Note

The Select Procedure dialog box displays the names of ALL procedures, whether they have been compiled or not, including those that are marked ***hidden****.*

Saving a Procedure

Save Procedure either saves a new Calibration Procedure or saves changes you made to the current one. The Save Procedure selection saves a Calibration Procedure to the Procedure Directory. You can save compiled or uncompiled procedures. If you save an uncompiled procedure, the procedure does not appear in the Procedure Selection dialog box in the Run Time application.

To save a procedure, first make sure the file you want to save is in the active window. Select **File**, then **Save Procedure**. This selection uses the procedure instrument name to save a procedure. In the status line of the Editor, a message appears indicating whether you saved an executable or non-executable procedure.

If a naming conflict arises, you are given the option of either overwriting the current name or giving the procedure a new name.

Opening a Text File

Open Text File lets you open any text file and view it.

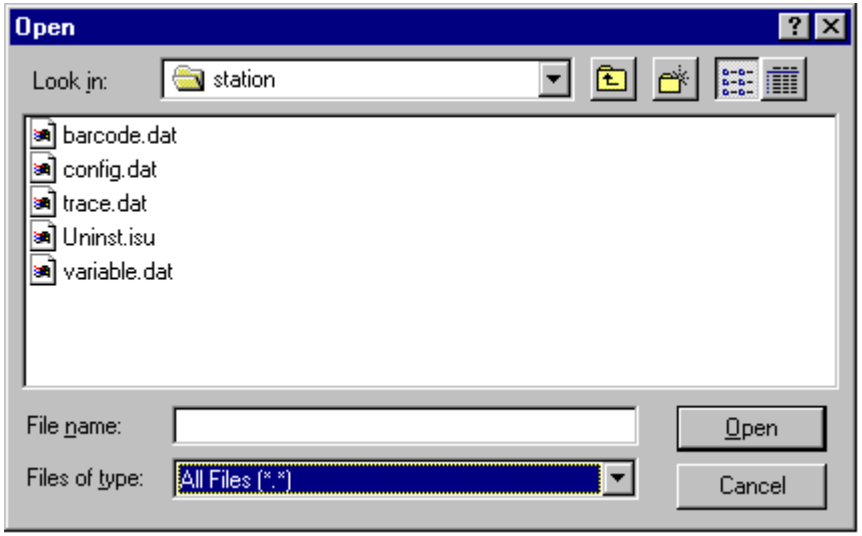
Note

If you do not perform Step 1 below, MET/CAL will attempt to compile the text file you load, because it assumes the text file is a procedure. Step 1 must be performed if you want to use the Editor to view or modify non-procedural text files.

To open a text file, proceed as follows:

1. From the menu, select **Configure**, then **Text Mode**.
2. From the menu, select **File**, then **Open Text File**.

   The Open dialog box appears.



zv604s.tif

Figure 6-4. Open Text File Dialog Box

3. Highlight the file name and select **Open**.

   The text of the file appears in the Editor window.

Saving a Text File

Save Text File lets you save the contents of the active window as a text file. Text files are free format and do not have special protection. Except where noted, case is not significant.

To save a text file, open the file and select **Save Text File**. A prompt appears if you have not yet specified a text file name for the procedure.

In the status line of the Editor, a message appears indicating the file is saved. The message indicates the name and location of the procedure.

Save As

Save As opens a dialog box that lets you save a new or changed procedure under a new name, directory, drive, and/or file type.

To save a procedure, proceed as follows:

1. Select the procedure you want to save.
2. From the File menu, click **Save As.**

   A dialog box opens.

3. Type in the new file name in the File Name field.

4. Select a directory to store the file and click **OK**.

   In the status line of the Editor, you get a message indicating the full pathname of the saved file.

Printing a Procedure

Print opens a dialog box that lets you print the current procedure. To print a procedure, proceed as follows:

1. Open the procedure you want to print.
2. From the File menu, select **Print**.

   A Print dialog box appears.
3. Choose the options you want and select **OK** to print the active procedure.

Print to File

Print to File lets you save the contents of the active window to a file with all format and printer codes. To print to a file, proceed as follows:

1. Open the procedure you want to print.
2. From the File menu, select **Print**.
3. Select Print to File, then **OK**.

   A Print to File dialog box appears.
4. Type in a name for the file to which you want to print. Enter a name different from the original to avoid overwriting the original file.
5. Select **OK**.

Print Setup Options

Print Setup lets you change the type of printer used, the page orientation, and the paper size. Depending on the printer available in your system, this selection has additional options that let you send the file as an encapsulated postscript file, remove margins, and create and send a header. Advanced options allow you to send particular true type fonts and manipulate graphics in your document.

Modifying Procedures

The Edit menu lets you modify procedures by selecting text within a procedure and choosing a command to make the desired changes. This section describes how to select and edit text within a procedure.

Using the Basic Edit Functions

The following paragraphs describe how to manipulate text within a procedure. These functions all start from the Edit menu.

Cut

Cut lets you cut selected text out of a procedure. Place your cursor at the beginning of the text you want to cut. Hold down the left mouse button and drag the cursor to highlight the text. Select **Cut**. The text is removed from the procedure and placed into the Windows clipboard, ready to be pasted.

Copy

Copy lets you copy selected text from one part of a written procedure to either another part of a procedure or a different procedure. To copy text, highlight the text you want to copy and select **Copy**. The text is copied into the Windows clipboard, ready to be pasted.

Copy to

Copy To lets you copy selected text to a file. Be sure this is what you want to do. If you have any data in the file you are copying to, it is overwritten.

Paste

Paste lets you insert the contents of the Windows clipboard at the current cursor position. The cursor may be positioned within any document window. To paste text after you copied or cut it, place the cursor where you want the text placed. From the Edit menu, click **Paste**. The text is positioned after the cursor.

Delete

Delete lets you delete selected data from a document. This selection is different than the Cut selection because once you delete information from a document, it cannot be recovered. To delete text, highlight the text you want removed and press **Delete**. Pressing **Delete** when the cursor is on a blank line deletes the line.

Break

Break lets you split a line in a procedure at the point the cursor is positioned and to start a new line. To break a line, place the cursor where you want the line to break and press Ctrl + ↵ Enter, or click **Break**.

Inserting a File

Insert File lets you insert the text of an external text file at the cursor position in the opened procedure. This is different from cut and paste or copy and paste.

When you select a file that you want to insert into the new procedure, **Insert File** copies and pastes everything in the source file. To insert a file, proceed as follows:

1. From the Edit menu, select **Insert File**.

   The Insert dialog box opens.

2. Select the file you want inserted into the active procedure, the select **OK**. The file is inserted into the active procedure and the dialog box closes.

Finding Text in a Procedure

Find lets you search for a text string. The editor searches forward or backward until the first occurrence of the text is found.

To find text, proceed as follows:

1. From the menu, select **Edit**, then **Find**.

   The Find dialog box appears.

2. In the **Find What** field, type in the text you are looking for.
3. Choose the options you want to use to control the search. You can specify the case sensitivity and direction (up or down in the procedure) of the search.
4. To start the search, click **Find Next**.

   If you find the selection you want to change, click **Cancel** and make your changes.

   To continue with the search, from the Edit menu, click **Find**. The Find dialog box remains available until you click **Cancel**.

   If the text for which you are searching occurs once in the procedure, the following message appears on the status line at the bottom of the screen:

   `Search string found at current position.`

Find Next

Find Next provides an alternate way to continue searching for text that you have specified in the Find dialog box. Use Find Next when you want to use the Find command without displaying the Find dialog box. You must have previously specified text under the Find option for this to work.

1. From the Edit menu, select **Find Next**.

   The found text is highlighted in the procedure, but a Find dialog box does not appear.

2. Select Find Next or F3 to continue searching.

Replacing Text in a Procedure

Replace lets you search for a specified text string and replaces each occurrence with different text. Replace lets you change text without performing cut and paste. To replace text, proceed as follows:

1. From the Edit menu, select **Replace**.

   The Replace dialog box appears.

2. In the **Find What** field, type in the text you want to find.

   If you used the Find or Replace options in the current session, the text from the previous search is already there. Type over the text to search for a different string.

3. Tab to the Replace With field and type in the replacement text.
4. Choose whether or not you want to match the case and select **Find Next** to start the search and replace operation.

   When a string is found, the following options become available:

   - **Find Next** goes on to the next occurrence of the string and does not perform a replacement.
   - **Replace** confirms a replacement.
   - **Replace All** replaces all strings without confirmation.
   - **Cancel** quits replace. The cursor remains where it was prior to quitting.

Adjustment Statement

Adjust indents a procedure line so the FSC is in the RANGE field. It is used only if the result of evaluation of the previous regular procedure step resulted in a MARGINAL or FAIL condition. For more information about Adjust, refer to the section titled “Adjustment Statement” in Chapter 1 of the Reference Manual.

Continue Statement

Continue lets you mark the current line as a continuation of the previous line. This is useful for certain FSCs that allow multi-line statements. For more information about Continue, refer to the section "Continuation" in Chapter 1 of the *Reference Manual*.

Naming a Procedure

Name Procedure specifies a procedure name. A procedure can have up to six names. The Editor keeps track of all names (instrument lines) and knows that all six names belong to one procedure.

To name a procedure, from the menu bar, select **Edit**, then **Name Procedure**. The Instrument: line appears. Type in the name you want for the procedure and press Enter. The name appears in the title bar of the active window.

For more information about naming a procedure, refer to "Naming a Procedure", in Chapter 1 of the Reference Manual.

Name Conflicts

A name conflict exists if any of the possible six names of one procedure is the same as a name in another procedure. These problems may occur when you modify or create new procedures with names already in the Procedure Directory.

Compiling Procedures

The Procedure Compiler is integrated with the Procedure Editor. The compiler checks the correctness of each procedure statement. Once the FSC is recognized, the compiler checks the correctness of individual fields within the line, and then the interdependencies of the lines.

The Compile menu displays selections that compile procedures forward and backwards, and individual lines of procedures. The Compile menu options are the following:

- Check line
- Next Error
- Previous Error
- Rewarn
- TUR Report

Checking a Procedure Line

Check Line checks the contents of the current line for errors, and reports errors in the error window. Errors must be corrected before you can run the procedure. If no errors are present, the line is formatted and fields are lined up under the correct columns. If the line contains an instrument evaluation FSC, the Test Uncertainty Ratio (T.U.R.) is calculated in most cases. If the T.U.R. is less than the "tur_lim" parameter specified in "metcal.ini," a warning message is generated and presented in the Error window. T.U.R. warnings are not considered errors, and so do not prevent a procedure from being executable. The procedure writer must verify that the new procedure is adequate, and that the standards used are satisfactory for the calibration to be performed.

Automatic Line Check is operating as long as the Editor is not operating in Text Mode. The same checking as described above is performed on every procedure line as you enter it. When you press [↵ Enter], the line check is triggered. If no errors are found, the line is formatted, and the cursor is placed on the next line ready for the next procedure statement.

Checking for the Next Error

Next Error lets you compile a procedure from the current line forward until an error is reached or until the entire procedure is compiled. If the last line is reached with no errors found, the compiler jumps to the top of the file and continues checking the procedure until the original line is reached or an error is found.

If the procedure contains no errors, the compiler checks for interline dependencies. For example, it checks for missing jump destinations, duplicate jump destinations, and incorrect indentation. If no compilation errors are found, the following message displays at the bottom of the screen:

No Errors. Procedure Executable.

If the procedure contains an error, an error window appears and indicates what the error is.

The cursor is positioned on the line containing the error in the procedure window. If the procedure window is not active, click the title bar to activate it.

Note

It is best to position the error window so that it does not overlap the procedure window. This allows the procedure window to remain active when the error window indicates an error.

Checking for a Previous Error

Previous Error is exactly the same as **Next Error**, except the procedure is scanned from the current line backward, rather than forward.

Rewarn

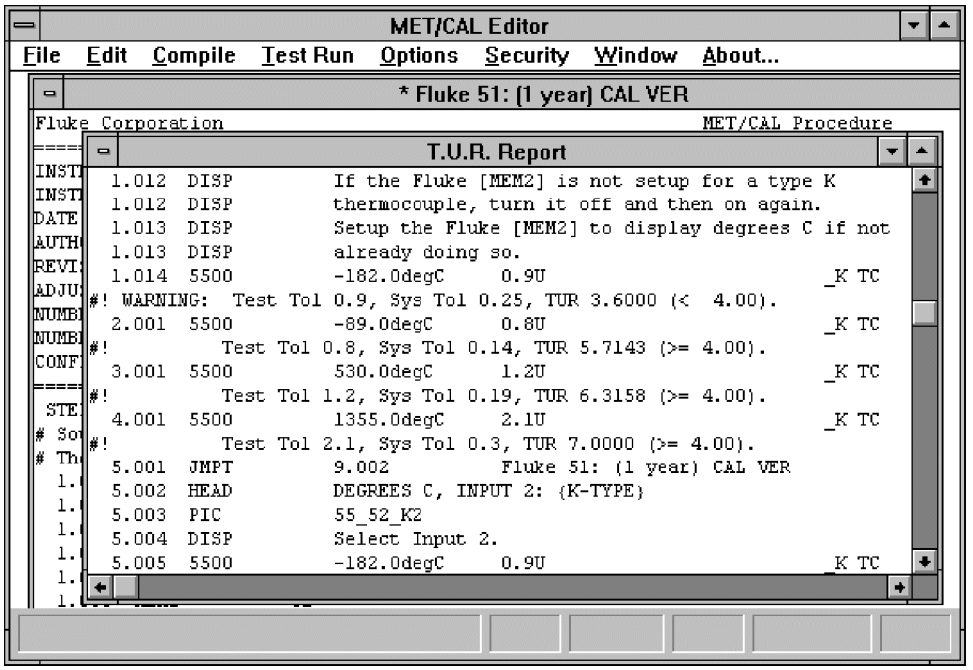
Normally, a Test Uncertainty Ratio (T.U.R.) warning is displayed only once. Subsequent compilation does not redisplay a warning unless you have modified the corresponding procedure statement. To override this, use Rewarn to enable all prior warnings in the active procedure.

T.U.R Report

T.U.R Report lets you create procedures with special comments that show the T.U.R. following each evaluation. A T.U.R. report helps you identify and address those tests where the application did not have enough information and either made an estimate or did not compute the T.U.R. These comments may indicate that the T.U.R. cannot be computed or was estimated based on range information. To generate a T.U.R. report, proceed as follows:

1. Open a procedure or use the one you are creating.
2. If the procedure is not compiled, compile the procedure. A T.U.R. report cannot be generated for a non-executable procedure.
3. From the Compile menu, select **T.U.R. Report**.

   A message indicates that the report is being generated. Then, a new window shows the procedure with the T.U.R. information embedded in it as comment lines. Refer to Figure 6-5.



il605s.tif

Figure 6-5. T.U.R. Report for the Fluke 51 Procedure

4. If you want to save the procedure with TUR information embedded, click Save Procedure.

   A message indicating that you have a name conflict appears. Click **OK**.

5. Another message appears asking if you want to overwrite the procedure that does not have TUR information embedded in it with the procedure that does. To overwrite, select **Yes**.

Note

When a new T.U.R.. report is generated, previously generated T.U.R. comments are stripped out.

Test Run Menu

Test Run lets you execute a procedure and to see how the procedure performs while it is being created or modified. You do not need to store a procedure in the Procedure Directory to test run it.

The procedure must be executable before you can perform a Test Run; if not, a warning message is shown asking you to check for errors. Check for errors using Next Error (F9).

When you test run, you are not asked for pre- and post-calibration information, and no results are written to the database. To test run a procedure, proceed as follows:

1. Select **Test Run** from the menu bar.

The following options are available:

- **Go**. Runs procedure from the line where the cursor is currently positioned Does not reset global procedure data.

When the procedure is completed or terminated, the values of certain parameters are remembered, so that the next time you execute the procedure, it is not necessary to repeat the test steps that establish those values.

For example, the ASK flags are typically defined close to the top of most procedures. You want to keep those settings even if you do not restart execution with the first line in the procedure. Remembering values provides the advantage of rerunning procedure segments with the test environment in the same condition each time. The first time you test run a procedure, start with those lines that assign values to those parameters.

The system remembers the following parameters:

- ASK flags, created using the ASK- and ASK+ FSCs.
- Memory variables MEM, MEM1, MEM2, and M1 through M255.
- Values created with Instrument Setup FSCs (For example, MMFC, MCAL, and M606 FSCs).
- MESS message buffers, created with the MESS FSC.
- DRAW parameters, created with the DRAW FSC.
- SET settings table, created with SET FSCs.

Configure Menu

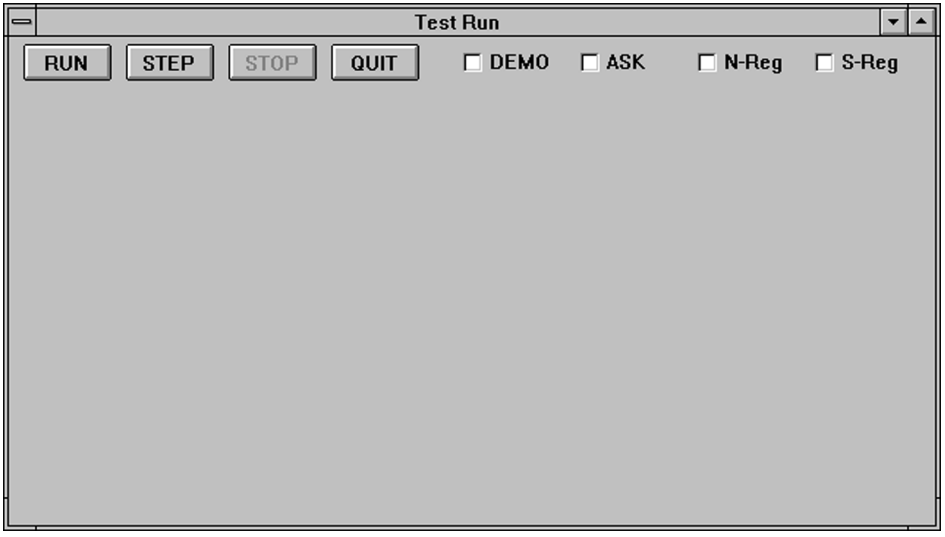
Selections on the Configure menu let you:

- switch between Text mode and Procedure mode
- choose the font used in displaying and printing the active Edit window
- configure the instruments that make up the workstation
- inform MET/CAL of number and type of IEEE-488 boards installed in the PC
- configure the defaults serial port for control of serial UUTs

- **Restart**. Runs a procedure from the top even if the cursor is not positioned at the beginning of the procedure. **Restart** also re-initializes global procedure data before the procedure runs.
- **Show**. Brings the Test Run window to the top to resume testing the procedure. While operating Test Run, you may need to switch to the Procedure window. This action covers the Test Run window.

1. Select one of the options.

The Test Run menu appears. Refer to Figure 6-6.



il606s.tif

Figure 6-6. Test Run Window

The following describes the function of the buttons and check boxes:

- **Run** executes a procedure. As the procedure runs, it halts only to prompt the user for information.
- **Step** executes the current statement and then pauses, waiting for the user to choose **Step**, **Run**, or **Quit**.
- **Stop** lets you stop execution when running a closed-loop procedure.
- **Quit** lets you exit Test Run and returns you to the Editor.
- **Demo** lets you run procedures without having instruments connected. Ordinarily, if a procedure attempts to communicate with an instrument, and no instrument is connected, the Editor generates a warning and the instrument will time out waiting for a response. Demo mode runs the procedure as if instruments actually are connected, but does not do a real I/O.

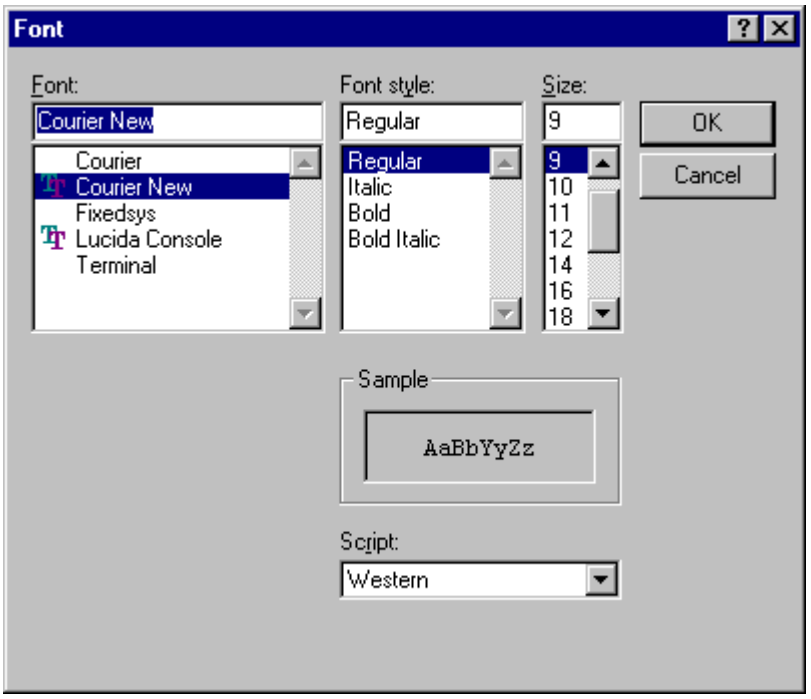
- **ASK** displays how the ASK flags are set in the procedure. ASK flags are control flags that affect automatic operation and/or messages. If you want brief description of what function each ASK flag performs, click the desired flag. The information about the selected ASK flag appears in the status line of the Editor. For more information about the ASK flags, refer to *ASK+* and *ASK-* in Chapter 2 of the Reference Manual.
- **N-Reg** shows the numeric registers and displays the values for MEM, MEM1, and M1-255 registers. For more information about MEM, MEM1, and M1-255, refer to the *MEME* and *MATH* FSCs in Chapter 2 of the Reference Manual.
- **S-Reg** shows the MEM2 string registers. For more information about MEM2, refer to *MEM2* in Chapter 2 of the Reference Manual.

2. After selecting a Test Run option, follow the instructions.
3. After you quit Test Run, you return to the Editor.

Text Mode lets you make changes to a text file other than a Calibration Procedure. In the Text mode, procedure syntax checking and compiling features are disabled. You can load, edit, and save text files. When you remove the check mark from the Text Mode menu item, full procedure checking is again available.

Font opens a Font dialog box and lets you make changes to the font, font style, and font size in the procedure that you are working on. To change a font, proceed as follows:

1. From the Configure menu, click **Font**. The Font dialog box opens. Refer to the following figure.



zv607s.bmp

Figure 6-7. Font Dialog Box

2. Select the font, font style, and size, and click **OK**. The font is changed only for the active edit window.

Instruments

The **Configure** menu provides a way to change the system instrument configuration. When you select **Instruments**, a small dialog box appears that contains a listing of all the currently configured instruments.

The following describes the function of the buttons:

Add Opens a second dialog box listing all the instrument FSCs available. From this, select the instrument you wish to add to the configuration. After you select one, you are immediately placed into the Edit Instrument dialog box, described next.

Edit Only available when one of the currently configured instruments is selected, (or opens automatically as a result of selecting an instrument to add). Opens a dialog box permitting port selection, alias assignment if desired, and specification of any options.

Delete Deletes an instrument from the configured instrument list.

Configuring Instruments

A configuration file maintains the list of configured instruments. Its default filename is CONFIG.DAT. Associated traceability information for configured instruments is stored in the database.

Before you can use calibration instruments, you must identify which instruments are configured in the system.

A configured instrument is a traceable calibration standard. There are three types of configured instruments:

System Instruments
A system instrument is an instrument for which the Fluke Metrology software provides a built-in instrument driver. For each system instrument there is a corresponding FSC in the procedure language. For example, the "5700" FSC is used to control the Fluke 5700A.

Fluke Metrology software supports over 30 different system instruments.

User-Configured Instruments
A user-configured instrument is one that is remotely controlled, but has no built-in driver. A procedure writer uses the PORT and IEEE FSCs to control user-configured instruments.

The PORT FSC is used for instruments with a serial interface (RS232) and the IEEE FSC is used for instruments with an IEEE-488 interface.

You do not need to make an instrument user-configured in order to communicate with the instrument using a Calibration Procedure. There are two advantages in using a user-configured instrument:

- A user-configured instrument can have an address and an alias. The alias is a symbolic name by which a procedure can refer to the instrument. Using an alias improves procedure readability and portability.
- A user-configured instrument is traceable and can be used in reports. Before starting a procedure that uses a user-configured instrument, the software first checks to make sure the instrument is in cal.

Also note that user-configured instrument specifications are subject to the same format and syntax rules as system devices, with the following additional restrictions:

- A user-configured device cannot have attached secondary devices.
- A user-configured device cannot have optional attribute specifiers.

Manual Instruments
A manual instrument is one that is not remotely controlled by the system, but is used as a calibration standard, so it must be traceable. Manual instruments can encompass a wide range of device types. Examples include the Fluke 792A

AC/DC Transfer Standard, the Fluke A40 Current Shunts, or a manual device such as a micrometer. The STD (Standard) FSC is used in a Calibration Procedure to require that an instrument be configured as a manual instrument standard. The DISP (Display) FSC is then used to inform the operator of setting or connection changes required when using the standard during execution of the calibration procedure.

Note

When configuring a new instrument, remember to also add traceability information to the database.

Modifying the Instrument Configuration

To add, delete, or change an instrument, the following steps are required:

1. Physically replace the instrument.
2. Edit the Configuration.

   To edit the Configuration, use the Editor application. Under the Configure Menu, select **Instruments**.
3. Update the configuration information.
4. Use the Manual Entry application to add or modify traceability information in the database.

IEEE-488 Boards

A dialog displays that allows you to specify the number (0, 1, or 2) and model of the IEEE-488 boards installed in your PC.

Default Serial Port

A dialog displays that allows you to specify the default port that the PORT FSC uses to control UUTs that have a serial interface.

Chapter 7
Customizing the Run Time Application

Contents **Page**

Introduction

This chapter describes how to customize an individual Calibration Workstation.

A number of editable files affect the Run Time environment. Changing them provides the ability to customize the operation of the MET/CAL and 5500/CAL Run Time applications. Table 7-1 lists the files that control the way the software operates.

Table 7-1. Overview of the MET/CAL Control Files

DEFAULT FILENAME	FUNCTION	SEE PAGE
RSLT_DB.FRM	Result Format File; defines which test parameters will be included in the detailed test results output that is sent to the database.	7-4
PROMPTDB.TXT PASS_DB.TXT FAIL_DB.TXT	Contain prompts to the user. The first set of prompts displays after selecting but before executing a calibration procedure. The PASS and FAIL prompt files display prompts upon completion of a procedure; which one the software uses depends on the result of the procedure.	7-12
CONNMESG.DAT	Assigns text strings to connection messages used by instrument drivers.	7-16
*.ACC	Accuracy Files	7-17
*.COR	Correction Files	7-24

Note

The filenames mentioned in this chapter are set in the initialization file, METCAL.INI. Except for the initialization file itself, these filenames can be changed by editing METCAL.INI.

Storing Test Results in the Database

To appreciate the reporting capability of Fluke Metrology software, you need an understanding of how results are stored in the database. With this information, you can select a report creation method to best fit your needs and available resources.

When a calibration procedure is run, the Run Time application stores results directly into the database tables. The saved calibration data contains three general types of information.

1. Test Summary
   Examples of summary data include UUT Identification, Cal Date, and Pass/Fail. This information is usually printed at the beginning of the report. Sometimes, relevant summary data is also included in page headers or footers.
2. Standards Used
   A list of all the standards used to perform the calibration.
3. Test Results
   The step-by-step test results of the calibration.

While summary and standards information are recorded into individual fields in the database, the step-by-step test results are handled differently.

Each evaluation (test step) in a MET/CAL of 5500/CAL Calibration Procedure generates a result. The Run Time application automatically writes the results to the database, as specified by a file called RSLT_DB.FRM (the results-to-database format file). This file is editable, so it is possible for the specific test results saved for any given calibration to differ from test results saved when RSLT_DB.FRM contained other specifications. The content of RSLT_DB.FRM at the time the calibration is performed controls the content and format of the test results saved to the database. This content also determines what information is available for later reports. When you run a report, the test results for the calibration of interest are returned.

Printing Issues

Each line of test results can be brought into any given report. Therefore, the format of report headers must match the format in effect at the time the test results were stored. The RSLT_DB.FRM file is made up of formatting instructions that describe what data to save, the order of the data items, and some additional

positioning information (where on the line to place a particular piece of information).

The result lines must be printed in a mono-spaced font, such as `Courier`. In this type of font, each character occupies the same amount of horizontal space.

The header lines you create for the report must contain appropriate titles, located at the same line position as the data below will be when the test results line comes out of the database. This is why it is important to understand the method by which formatted test results are stored.

Take particular care if you want to modify RSLT_DB.FRM. Results saved to the database are determined by the contents of this file, and, if you modify it, any reports you create must match the settings in the RSLT_DB.FRM file that was in effect at the time the test results were stored. If you save results for a series of calibrations, and then you edit RSLT_DB.FRM, be sure to save the original report definitions (*filename*.rpt) so that you can retrieve older results data later. This is easily accomplished by modifying the report in conformity with format file changes, but saving the new report under a new filename.

Edit reports and the RSLT_DB.FRM file in tandem.

For example, if you decide to not include one of the variables mentioned in the results-to-database formatting file, you should create new reports to reflect that change. Otherwise, you end up with report headers that do not match the data returned. Anytime you want to edit either the RSLT_DB.FRM file or a report based on test results formatted using that file, **edit both files**. Otherwise, report headings may not match the stored test results data.

Later in this section, an example graphically shows the connection between the RSLT_DB.FRM file and the stored test results.

The reports supplied with the Fluke Metrology software were created to conform to the RSLT_DB.FRM file as shipped. If you modify either a supplied report or the RSLT_DB.FRM file, make the corresponding changes to the other file.

Format of Test Result Lines (RSLT_DB.FRM)

The Result Format file **`RSLT_DB.FRM`** defines which test parameters will be included in the detailed test results output that is sent to the database. When you later run a report asking for information about that calibration, the formatted test results line is returned. Note that the entire formatted test results line is stored in the database. It is not possible to select and format only certain elements of the test results. Your report can only display the entire formatted test results line.

_Contents of RSLT_DB.FRM_

The RSLT_DB.FRM file can contain three functions. Examples for using each of the functions can be found later in this chapter.

RSLT_TYPE$() Takes two arguments: the result type and variables. Together, these arguments define the type and content of each test results line.

SPLIT$() Some test results lines automatically include introductory text. For example, a result of using the System Tolerance variable ST$ might read:

```
SYSTEM TOL:9mV
```

You can use the SPLIT$() function to strip off everything except the returned value, so the stored result would be:

9mV

SUBST$() Test results lines are stored as ASCII text, so there are no symbols available for “ohms”, “micro”, and other symbols. Instead, the stored results ordinarily employ little-used characters. For example “Z” is used instead of “ohms.” The SUBST$() function lets you substitute the saved value for a more meaningful string in the output. For example, you can convert all upper-case “Z” characters to the word “ohms.”

_The RSLT_TYPE$() Function_

Each format line in the **RSLT_DB.FRM** starts with the following:

```
RSLT_TYPE$(...)
```

The RSLT_TYPE$() function in the **RSLT_DB.FRM** file selects the appropriate formatting for each type of result output.

Each format line starts with “RSLT_TYPE$(...)” and ends with the next “RSLT_TYPE$(...)” entry. A blank line in the format file causes a blank line to be added to the test results output that is sent to the database.

List of Result Types

The argument to the RSLT_TYPE$() function defines the type of result to be created. There are eleven result types, as described in the following table. Each result type can take a number of variables. These are shown in the rightmost column. The variables are defined in a table following this one.

Table 7-2. Result Types That Can Appear in RSLT_DB.FRM

Result Type	Meaning	Variables (Details in Next Table)
INSTR	The normal output of an instrument FSC.	TN$ UR$ UI$ SA$ MO$ ER$ ET$ ST$ TT$ TU$ CO$ TR$ LL$ UL$ FN$ FVA$ FRE$ FUL$ FLL$ A$
ADJUST	Relates to the "Adjustment Performed" result output.	TN$ RM$
BRACE	Output created by enclosing text in braces in the procedure listing.	TN$ RM$
DOSE-NUM	Numerical type of result output created with the DOSE FSC.	TN$ RM$ ET$ CO$ TR$
DOSE-PF	PASS/FAIL type of result output created with the DOSE FSC.	TN$ RM$ TR$
EVAL	Result output created using the EVAL FSC.	TN$ RM$ TR$
GONOGO	Go/No-Go output from an instrument FSC.	TN$ UR$ UI$ SA$ MO$ TR$ ST$ TT$ TU$ CO$ LL$ UL$ FN$ FVA$ FRE$ FUL$ FLL$ A$
REMARK	The output created when an operator enters a remark by selecting REM in the Post-Test summary.	TN$ RM$
RSLT	Output created using the RSLT FSC in a Calibration Procedure.	TN$ RM$
SEPARATOR	Defines the separator between tests. Default is a blank line. If no blank line follows the result type specifier, the Run Time will not include blank lines in the output.	
END	Defines the last entry in the result output after execution of the Calibration Procedure.	

Variables in the Results Format File

Each result type that can appear in the RSLT_DB.FRM file can take a number of variables. Variables mentioned in the format file are substituted for the value of the data when the test results are stored, as long as the variable exists in the RSLT_DB.FRM file at the time the calibration is performed.

The following table describes the values that are substituted into a formatted test results line for each of the possible variables.

Table 7-3. Result Type Variables

Variable	Meaning
A$	Note flag. A$ is used to indicate that FRE$ represents a System_Actual value, rather than UUT_Indicated. The result of this flag is either an asterisk or a blank space, depending on the type of test. An asterisk may refer to a comment that explains its usage.
CO$	Comments. Contains one of the following: • If the result of the test is between the adjustment threshold and 100% of tolerance, CO$ contains: MARGINAL • If the test was executed the first time and the result exceeded the limit set in the metcal.ini file as defined by the entry: oot_lim = 100% , CO$ contains: NOTIFY USER. 100% is the default value for the Out-Of-Tolerance Limit. • CO$ is blank in all other cases
DE$	Error in units of the nominal.
ER$	Relative Error in % (if ≥0.1%) or PPM (if <1000PPM). If nominal is zero or expressed in dBm, ER$ is expressed as an absolute value.
ERPPM$	Error in ppm (or absolute). Similar to ER$, except that ERPPM$ does not switch between percent and ppm and uses slightly wider fields than ER$.
ERPCT$	Error in percent (or absolute). Similar to ER$, except that ERPCT$ does not switch between percent and ppm and uses slightly wider fields than ER$.
ET$	Error as percent of Tolerance.
FLL$	Formatted Lower Limit. The FLL$ is the allowed lower limit for FRE$. The units are assumed to be identical to those in FN$.
FN$	Formatted Nominal and MOD1 value (e.g., 10.000V @ 100 kHz)

Table 5-2. Result Type Variables (cont.)

FRE$	Formatted Result value. FRE$ contains the result value. Depending on the type of test, it matches either UUT_Indicated or System_Actual (see detailed explanation later). The units are assumed to be identical to those in FN$.
FVA$	Formatted Reference value. FVA$ contains the true nominal stimulus output if different from the Nominal value. This occurs, for example, with Ohms stimulus, and when corrections are applied. If the value of FN$ is the same as FVA$, FVA$ is left blank. No units are shown; units are assumed to be identical to those in FN$.
FUL$	Formatted Upper Limit. FUL$ is the allowed upper limit for FRE$. The units are assumed to be identical to those in FN$.
LL$	Lower Test Limit.
MO$	Modifier value (frequency most often). Maximum of 14 characters.
RM$	Remark. Contains the text created by certain non-instrument FSCs. The following examples indicate what text will appear in the output: • DOSE FSC: up to 64 characters can be generated. • EVAL FSC: text created by the EVAL FSC. • RSLT FSC: text shown in the listing of the procedure. • Adjust message: “ADJUST PERFORMED” • {Braced Data) in the procedure listing. • Operator entered remark: up to 32 characters can be entered.
SA$	System Actual. The actual calibrator output at the time the test ran (if the UUT is a measurement device) or the actual value measured by the calibration instrument (if the UUT is a stimulus device). This value is determined by an accurate calibration instrument in your system. Maximum 14 characters. The string “OPER-EVAL” indicates a GO/GO-NO status message.
ST$	System Tolerance. May be blank if there is no Test Uncertainty Ratio Limit checking, as signified by this entry in the **metcal.ini** file: tur_lim = none. See description following this table.
TN$	The test or subtest number. The range is from 1 to 999. The highest test number is 999, and the highest subtest number is 999.999.
MU_COV$	Coverage factor used in measurement uncertainty calculation.
MU_EXP$	Standard measurement uncertainty.
MU_STD$	Expanded measurement uncertainty.

Table 5-2. Result Type Variables (cont.)

TR$	Test Result. Contains either PASS or FAIL.
TT$	Test Tolerance. May be blank if there is no Test Uncertainty Ratio Limit checking, as signified by this entry in the **metcal.ini** file: tur_lim = none. An entry is created only if the Test Uncertainty Ratio (TUR) is below the limit defined in the **metcal.ini** file (tur_lim = 4 as a default). If there is no warning, the entry is blank. See description following this table.
TU$	Test Uncertainty Ratio of the test. An entry is created only if the Test Uncertainty Ratio (TUR) is below the limit defined in the **metcal.ini** file (tur_lim = 4 as a default). If there is no warning, the entry is blank.
UI$	UUT Indicated. The value the UUT indicated as a result of the test. Maximum of 14 characters.
UL$	Upper Test Limit.
UR$	UUT Range. Maximum of 5 characters. If blank, no range is specified. The single letter 'A' indicates autoranging.

The SPLIT$() Function

The Split function allows you to remove certain text that would otherwise appear automatically in the output. It splits a variable into several fields. The format for using SPLIT$() is as follows:

`SPLIT$(<initial-var>,<var1>,<match-chars1>, <var2>,<match-chars2>, etc.)`

where:

`<initial-var>` The string of characters to be split. The leading and trailing blanks are removed before performing the split.

`<var1>` A variable that is assigned to the characters up to the first matching character as defined in `<match-chars1>`.

`<match-chars1>` The sequence of matching characters that signifies the termination of `<var1>`. Again, leading and trailing spaces are removed.

The above process of matching is repeated for `<var2>` and `<match-chars2>`, etc.

For example, when the Run Time application generates a result line that includes the ST$ variable, the result line normally contains the text:

```
SYSTEM TOL:9mV
```

You can strip "SYSTEM TOL:" out of the result line by using the SPLIT$() function. To use this function, make an entry like the following in the RSLT_DB.FRM file:

```
SPLIT$(ST$,ST$,":",ST$)
```

Examples Showing How to Use the Split Function

- **TT$ Test Tolerance**
  The TT$ variable normally contains the text:

  ```
  TEST TOL:9mV
  ```

  Strip "TEST TOL:" from the variable TT$ using the SPLIT$() function as shown below as an entry in the **RSLT.FRM** file:

  ```
  SPLIT$(TT$,TT$,":",TT$)
  ```

- **TU$ Test Uncertainty Ratio Warning**
  An entry is created only if the Test Uncertainty Ratio (TUR) is below the limit defined in the **metcal.ini** file (tur_lim = 4 as a default). If there is no warning, the entry is blank.
  Strip "TUR:" from the variable TU$ using the SPLIT$() function as shown below as an entry in the **RSLT.FRM** file:

  ```
  SPLIT$(TU$,TU$,":",TU$)
  ```

The SUBST$() Function

When a Calibration Procedure produces test results lines, they contain special symbols, so they may not be very readable. One example involves the ohms symbol. The raw result output uses the upper-case letter Z to indicate the word, but usually, you want the test results output to indicate either the word ohms, or to use the Greek *omega* symbol, Ω. To convert these placeholder characters to something meaningful, use the subst$() function.

Example: to substitute the string nZ to the string nOhm (nano-ohms) in the $FN position, use the subst$() function as follows:

```
subst$ (FN$,"nZ"," nOhm")
```

In practice, you rarely need to write such conversion functions, since all those required are already present in the supplied RSLT_DB.FRM file.

Additional Rules for the RSLT_DB.FRM File

The following additional rules apply to the RSLT_DB.FRM file:

- Only defined variables can be used in the **RSLT_DB.FRM** file.

  The Run Time application substitutes the value for the variable appearing in the **RSLT_DB.FRM** file.

- Lines that have a pound sign (**#**) as the first non-blank character are considered to be comments and are ignored.
- Arbitrary ASCII text is copied exactly to the test results output.
- Blank lines are significant. They are included in the results output.
- A backslash (****) in front of a character causes that character to be placed literally in the test results output.

  The backslash can be used for any character including the backslash character itself – use two backslashes (****) to place a single one in the result output.

  This ensures that you can obtain proper output even if you are unsure whether a character is somehow special. If in doubt, add the backslash.

  Example: **\%** would place a **%** symbol in the result output.

- Even though the dollar sign (**$**) is a special character in the context of variable names, it is not when it appears in a text context. In that case, a dollar sign by itself is placed directly into the result output. There is no need to precede it with a backslash.

Customizing Prompts for the Operator (PROMPTDB.TXT, PASS_DB.TXT, FAIL_DB.TXT)

You must edit the **PROMPTDB.TXT, PASS_DB.TXT,** and **FAIL_DB.TXT** files to collect customized information that you want to have included in any type of results printout. You can edit the prompt files using the Procedure Editor's "Open Text File" selection (File Menu) or using an external text editor.

The Run Time application prompts the operator of the calibration system every time a calibration procedure is executed, if the system has been so configured. Prompts occur in the following instances:

- Pre-run prompt (PROMPTDB.TXT): After selecting a procedure to be executed, but before a procedure actually starts up.
- Post-run, Pass (PASS_DB.TXT): After completion of a procedure, and the UUT passes.
- Post-run, Fail (FAIL_DB.TXT): After completion of a procedure, and the UUT fails.

The presence of appropriate prompt files determines if one or more of these prompts occurs, and their content determines the type of information for which the operator is prompted.

When the operator enters a response, the text entered is assigned to the name of a variable. The variable and the assigned string value are stored in a temporary variable file. The values can be recalled when results output is created. The temporary file is not erased after a procedure terminates; values can be "recalled" automatically when starting up another procedure, using special constructs in the prompt files.

The prompt at the beginning of execution of a calibration procedure is determined by the prompt file with default name **PROMPTDB.TXT**. The default content of the **PROMPTDB.TXT** file is shown below:

```
Temperature in deg C      ::  TEMP$  2311
Relative Humidity in %    ::  RH$    2312
Calibration Seals Intact? :B: SEALS$ 2322
Work Order Number         :B: WO$    2314
```

Prompt File Description

Each line in a prompt file specifies a prompt to the user that appears on a Run Time prompt screen. Except for blank lines and comment lines, both of which are ignored, each line in a prompt file has 3 or 4 components as follows:

Prompt String The prompt string is the text that the user sees when the prompt screen appears. The prompt string is everything on the line up to the first colon (:). The colon is not included in the prompt string, but it must appear in the prompt file, otherwise there is an error.

Attribute String The attribute string includes everything between the first and second colons. If there is no second colon, the attribute string is just the colon.

The attribute string can contain the following specifiers:

R Indicates that the field is read-only

N Indicates the field must contain a numeric value only

B Indicates the response buffer is to be initially blank. An attribute specifier of **:B:** is equivalent to just a colon - **:**

However, **B** can be useful in conjunction with other specifiers. For example, **:BN:** specifies an initially blank, but numeric-only field. Note that this is not the same as **:N:**, which specifies a numeric-only field initialized with the value from the variable file.

<n> An integer between 1 and 128 restricts the maximum number of characters that can be entered in the response field.

D Indicates that the response buffer is to be initialized with the *default* database value (NOT the current value of the field). Permitted fields: In Version 6.0, Inventory Table fields are allowed if they are read-only, calibration table fields, and those that are not updated automatically. Specifically, the following Calibration table fields cannot be prompted for:

2301	2319	2333	2339	2346	2351
2305	2320	2335	2342	2347	2352
2315	2321	2336	2343	2348	2353
2317	2323	2337	2344	2349	2363
2318	2325	2338	2345	2350	

Variable Name String The variable name string is the name of the variable that will be associated with the user's response.

The variable name string is everything after the attribute string up to the terminating dollar sign ($). The dollar sign is required. Leading white space is not included in the variable name string.

Database Field Name String The database field name string is optional. If it is included, it specifies the field in the Calibration table of the database to which the user's response will be written.

The database field name string is anything after the dollar sign. It does not include any leading or trailing white space. It may be empty (since the string is optional).

The name can be a field number, or it can include a leading "C" to designate Calibration table fields.

Prompt File Rules

The following rules apply to the Prompt File:

- Every line in the prompt file represents one prompt.
- The <prompt> can be any character except colon (:), and is used to provide the prompt for the operator.
- A <variable-name> must start with an alpha character (a-z, A-Z) or underscore (_), and is optionally followed by more alphanumeric characters or the underscore "_" character.
- A <variable-name> must end with a dollar ($) sign.
- Upper case and lower case is not significant in variable names.

- If the colon is replaced by two colons, the entry is preloaded with the currently assigned value stored in the **VARIABLE.DAT** file.

  For example, suppose the temperature and relative humidity in your calibration laboratory are closely controlled. The temperature and %RH are not expected to change significantly. Also, the name of your calibration facility is not changing. The previously entered value is not preloaded if the entry in the prompt file has a single colon:

  ```
  Temperature          : TEMP$
  ```

  The operator can override the preloaded entry, speeding up the operation by not requiring new entries for values that seldom change.

  The variable name and its value is stored in the **VARIABLE.DAT** file. The variable name and value can be accessed by programs run through the DOS and DOSE FSCs, and can be inserted into the results via the **RSLT.FRM** file.

Parsing and Validation of Attribute Strings

The following rules control the parsing and validation of the attribute string:

1. White space is ignored, except that white space cannot be embedded within the maximum response length specifier (<n>).
2. No specifier can be used more than once.
3. Specifiers can occur in any order.
4. The read-only specifier (R) is mutually exclusive with all other specifiers, except the “use database” specifier (D).
5. Whenever the attribute string contains two colons, the prompt response buffer will be initialized, unless the Blank specifier (B) appears.

   The initialization value is taken either from the variable file or from the database default value for the field. The database default is used only if (a) the prompt file line specifies a database destination field, and (b) the attribute specification includes the **D** specifier.
6. If the attribute string is simply a colon (:), no attribute specifiers are present. In this case, the attribute is really just separating the prompt string from the variable name. The response buffer displayed to the user will initially be blank.

Prompt Fonts

You can select the font to use for prompt screens through the initialization files, METCAL.INI. There are three possible selections: SYSTEM, DISPLAY, and PROMPT. For details, refer to the heading earlier in this section that describes initialization file entries.

Post-Run Prompts and the As Found / As Left Feature

The purpose of field 2308, Found/Left, is to permit you to mark each calibration event with a tag specifying the point in your process that the event occurred. The tags specify whether the calibration results indicate the condition of the instrument at the time it arrived, at the time it left, or both.

The As Found / As Left feature makes use of prompts contained in both the FAIL_DB.TXT and PASS_DB.TXT files. The prompts create a validated field on the Pass and Fail Post Run prompt screens (assuming that the validation entries for 2308 have not been modified).

Refer to Chapter 8, “Configuring the System” for complete information about how to use the As Found / As Left feature. There, you will find a discussion that explains how to use the feature in conjunction with the Pass/Fail field to create a reliability profile for your assets.

Connection Message File (CONNMESG.DAT)

The **CONNMESG.DAT** connection message file contains the text of connection messages automatically generated by the instrument drivers in the Run Time application and Editor Test Run.

The **CONNMESG.DAT** file is an ASCII file that consists of a series of lines, each containing a connection message number followed by a connection message. Users may modify the text of existing connection messages but may NOT renumber existing messages.

Instrument drivers refer to connection messages by number. Each time a connection message is needed, the file that contains the connection messages is accessed to obtain the actual message to be displayed.

You can use the default **CONNMESG.DAT** file as is, independent of the calibration system and controller configurations, unless a special instrument is added in your system.

Description of Accuracy Files (*.ACC)

Unless TUR checking is turned off, there must be one Accuracy File associated with each Instrument FSC. Accuracy files contain information on the specified accuracy of the instrument associated with the FSC.

Accuracy files may be empty. In this case, the Test Uncertainty Ratio checking must be disabled by the following entry in the **metcal.ini** file:

```
tur_lim = none
```

The Procedure Compiler verifies that the requested value of parameters as stated in the FSC is possible. It also attempts to compute calibration accuracy ratios and issue a warning if the ratio of specified UUT tolerance to tolerance of the calibration system at the Nominal value drops below a constant value. This constant value is the one assigned to the variable tur_lim in the **metcal.ini** file. The default value is 4.

The person who is writing your procedures should ensure that TUR warnings are not generated by verification steps.

However, during adjustment steps, warnings are commonplace and generally can be ignored. The reason is that calibration adjustments are typically made close to the resolution of the UUT, which can be well within the stated accuracy of the UUT. In the limit case, the reading is set right on, meaning zero tolerance. Given that any calibration instrument has at least some tolerance, a warning of insufficient Test Uncertainty Ratio is generated that can be ignored in such a case.

The TUR is defined as (Test Tolerance / System Accuracy).

The compile time and run time TUR calculations may not be identical. At compile time the Test Tolerance is determined as the distance from the specified Nominal to the closest tolerance limit, unless one of the limits is zero. For example, a +3U in the tolerance field (implying -0U tolerance) causes the Test Tolerance to be 3U.

At run time, the Test Tolerance is calculated as the distance from the UUT Indicated value to the tolerance limit that falls on the SAME side of the UUT Indicated value as the System Actual.

All accuracy files are located in the directory specified by the “accdir” parameter in the **metcal.ini** file. Accuracy files are named as follows:

```
{FSC}.ACC
```

Example:

```
5100.ACC
```

Every Fluke calibration software package includes default accuracy files for all instruments in the system. These accuracy files reflect the performance of those calibration instruments in a systems environment over a period of 90 days. In most cases, no changes are needed.

Certain calibration instruments may have special calibrations, making it possible to achieve tighter specifications. By modifying the accuracy files, it is possible to inform the calibration system of these tighter specifications.

Note that the ACCF FSC modifies how the software handles accuracy files. For details, refer to the description of the ACCF FSC in the Procedure Reference manual supplement.

Accuracy File Formats

- Accuracy files are ASCII files that can be modified using any text editor.
- Each accuracy file must begin with a file header.
- Text is case insensitive.
- The file format is as follows. Each file is a sequence of sections. Each section consists of a MODE line, followed by one or more lines containing numeric data.

  A MODE line has the format:

  ```
  MODE: <unit specifier> [ <flag> <flag> ....]
  ```

- The list of valid unit specifiers depends on the instrument. For example, valid unit specifiers for the 5100 FSC are volts, decibels, and amps.
- Flags are also specific to each instrument. For the same 5100 FSC, valid flags are E (for accuracy enhancement) and O (for Divider Override).
- A mode line may not have extraneous characters.
- Mode lines are not user-configurable.
- Numeric data lines containing the accuracy information for the stated MODE have the following format:

  <nlb> <nub> <mlb> <mub> <tol> <floor> <res>

  where:

  <nlb> is the nominal lower bound
  <nub> is the nominal upper bound
  <mlb> is the MOD1 lower bound
  <mub> is the MOD1 upper bound

<tol> is the tolerance as a percentage of the nominal value (%)

<flr> is the tolerance in absolute units (floor specification)

<res> is the resolution of the instrument in absolute units or percent of nominal

- The string “NA” may be used to indicate that a particular value is not applicable (i.e., not used).
- The nominal bounds must always be specified.
- The lower and upper bounds for Nominal and MOD1 are the legal lowest and highest values at which the instrument can be programmed. The lower and upper bounds themselves are legal values.
- The tolerance and floor may both contain “NA” to indicate that a range is valid, but no accuracy is specified for that range.
- If either tolerance or floor is “NA”, both must be “NA.”
- The resolution is expressed either in absolute units or percent of the Nominal value. If expressed in percent, the resolution is terminated with the percent (%) sign. Resolution may be “NA.”
- The file may contain any number of blank lines or comment lines.
- Comment lines have a pound sign (#) as the first non-white space character.
- White space before the beginning of text or after the text of a line is not significant.

A section of the accuracy file for the 5100 FSC may appear as follows:

```
Mode: Volts

0   .0199999  NA  NA  0.0038  0.0000052 0.1e-6
.02 .199999   NA  NA  0.0038  0.0000066 1e-6
.2  1.99999   NA  NA  0.0038  0.000021  .00001
2   19.9999   NA  NA  0.0038  0.000165  .0001
20  199.999   NA  NA  0.0038  0.0016    .001
200 1100      NA  NA  0.0038  0.016     .01

# Base 5100B accuracy AC Volts with divider for
# low values < 20 milliVolts.

1e-6 .0199999  50    10e3  0.1   0.00001  0.1e-6
1e-6 .0199999  20e3  50e3  0.12  0.00001  0.1e-6
.02  .199999   50    10e3  0.05  0.00006  1e-6
.02  .199999   20e3  50e3  0.08  0.000066 1e-6
.2   1.99999   50    0.9e3 0.05  0.00015  .00001
.2   1.99999   1e3   10e3  0.05  0.00013  .00001
.2   1.99999   20e3  50e3  0.08  0.00021  .00001
2    19.9999   50    10e3  0.05  0.001    .0001
2    19.9999   20e3  50e3  0.08  0.00165  .0001
20   110       50    10e3  0.05  0.01     .001
20   110       20e3  20e3  0.08  0.016    .001
110  199.999   50    1e3   0.05  0.01     .001
200  1100      50    1e3   0.05  0.055    .01

Mode: Volts E
# Enhanced accuracy specification for 7404B
# type system
# The accuracy enhancement is done with the 8506A from
# the output directly.

0    .0199999  NA   NA   0.0022   0.0000016   0.1e-6
.02  .199999   NA   NA   0.0022   0.0000016   1e-6
.2   1.99999   NA   NA   0.0014   0.0000062   .00001
2    19.9999   NA   NA   0.0010   0.000160    .0001
20   127.999   NA   NA   0.0017   0.0006      .001
128  199.999   NA   NA   0.0017   0.006       .001
200  1100      NA   NA   0.0017   0.006       .01
```

Accuracy File Headers

Beginning with Version 6, each MET/CAL accuracy file must have a header. All accuracy files that are shipped with Version 6 will include valid headers. If you have created your own accuracy files you will have to add headers to these files.

Accuracy file headers allow MET/CAL to automatically associate the correct accuracy specifications with an instrument that is used as a calibration standard.

As in previous versions of MET/CAL, accuracy files are used only with supported system instruments. These are remotely-controlled calibration standards for which MET/CAL includes a built-in driver.

Examples of supported system instruments are the Fluke 5700A, Fluke 5500A, and the HP 3458A. Accuracy files are not used with user-configured instruments or with manual standards.

The accuracy file header should appear at the beginning of the accuracy file. Comments may precede the header.

The accuracy file header is used to specify the instrument name, calibration interval, confidence and asset.

Instrument Name

This is the system instrument name that is associated with the accuracy file. The name must agree exactly with the name of a supported system instrument. For example, Fluke 5700A is valid, but Fluke 5700 is not.

The syntax for this parameter is:

```
instrument = <name>
```

Name recognition is case-insensitive.

The accuracy file header must include an instrument specification.

Calibration Interval

This is the calibration interval associated with the accuracy file. For example, if the file contains accuracies based on 90-day specs for a 5700A, the interval should be 90 days.

The syntax for this parameter is:

```
interval = <interval number><interval units>
```

The `<interval number>` may be any number. Units may be specified in days, weeks, or months. For days, `<interval units>` may be:

"d", "da", "day", or "days"

For weeks, `<interval units>` may be:

"w", "wk", "week", or "weeks"

For months, `<interval units>` may be:

"m", "mo", "mon", "month", or "months"

Calibration nterval recognition is case-insensitive.

The accuracy file header must include a calibration interval specification.

When MET/CAL compares the calibration interval that is specified in an accuracy file header to the calibration interval associated with an asset in the database, it considers the two intervals to be the same if they are within approximately 10% of one another. This method allows, for example, 365 days to be recognized as 12 months.

Confidence

The confidence parameter specifies a sigma value for use in the measurement uncertainty calculation. The syntax for this parameter is:

```
confidence = <confidence value> [sigma]
```

The confidence must be specified as a sigma value. It cannot be specified as a percentage. The word sigma is optional in the specification. If specified, recognition is case-insensitive.

The accuracy file header must include a confidence specification.

Asset

The asset parameter specifies an asset number that is associated with the accuracy file. The asset specification is optional. If specified, it indicates that the file should be used only with the specified asset.

The syntax for this parameter is:

```
asset = <asset number>
```

If specified, the `<asset number>` should be the same one entered in the database for the calibration asset.

Sample Accuracy File Headers

Each accuracy file header is delimited by the lines:

```
Begin Header
End Header
```

Header delimiter line recognition is **not** case sensitive.

An example of a complete accuracy file header is:

```
Begin Header
instrument = Fluke 5700A
interval   = 90 days
confidence = 2.58 sigma
End Header
```

An example of a the same header with an asset number specification is:

```
Begin Header
instrument = Fluke 5700A
interval   = 90 days
confidence = 2.58 sigma
asset      = 23-495A
End Header
```

Adding the asset specification in the example indicates that this accuracy file can be used only with asset 23-495A.

Avoid Accuracy File Header Conflicts

It is important to avoid accuracy file header conflicts. Two accuracy files in the MET/CAL accuracy directory may not specify the same instrument name and same calibration interval, unless you have included unique asset number specifications in one or both headers. In addition, if the procedure contains ACCF statements that refer to accuracy files not in the MET/CAL accuracy directory, the headers of those files must not conflict with the headers of any file in the MET/CAL accuracy directory, nor with the header of any other ACCF-specified file in the same procedure, or any called subprocedure.

The confidence specification in the accuracy file header cannot be used to distinguish two accuracy files which otherwise have the same header. This is because, although the confidence value is used in the measurement uncertainty calculation, it is not one of the selection criteria used to automatically associate an accuracy file to a configured calibration asset.

Creating Correction Files (.COR)*

One Correction File may be associated with any Instrument FSC. These files are optional. If used, they are located in the directory specified by the "correctdir=" parameter in the **metcal.ini** file. Correction files must have file names ending with **.COR** . Correction files contain information on special calibration points of the instrument associated with the instrument FSC. An electronic instrument is far more stable during a limited period of time than its stated accuracy indicates. By providing specific calibration points and reducing the calibration interval, the uncertainty of the calibration instrument under those conditions can be significantly reduced. This is the case in particular with AC voltage functions.

The presence of correction factors is indicated by the value of the Correction File Directory parameter "correctdir" in the **metcal.ini** file being set to something other than "none." The default is "none." If the Correction File Directory is identified, a search for correction factors is done. If no correction factors are known, the files can be blank.

If there is a directory named **CORRECT**, the Run Time application attempts to find the corrected value for the calibration instrument, and report this corrected value in the results.

Note

If Correction files are used, the Accuracy file and Calibration Interval should also be changed to reflect the improved accuracy specifications.

To indicate that a correction has been applied, the software displays the corrected System Actual value in the Post Test Summary dialog, followed by the word **(corrected)** to the right of the System Actual value.

The Test Results window also indicates corrections. The corrected System Actual is shown in the *System Actual* column, and a **Y** appears in the *C* column.

You can access the correction indication in the formatted results using the C$ variable.

How Correction Files are Found and Named

The **correctdir** directory contains all correction files. Correction files are named as follows:

```
<FSC>.COR
```

Example:

```
5100.COR.
```

The Fluke Metrology software package includes no default correction files. An example correction file for the 5200A is illustrated later in this chapter.

Interpolation Methods to Determine Corrections

The Correction file first identifies the modes for which corrections are available. For example, the corrections apply to the volts mode:

```
Mode: Volts
```

Any qualifiers, such as "O", are included. This is also the case with the mode information in the accuracy files:

```
Mode: Volts O
```

Within each mode, a number of ranges can be defined over which interpolations can be made. This includes the following:

- A lower bound for NOMINAL (e.g., voltage)
- An upper bound for NOMINAL (e.g., voltage)
- A lower bound for MOD1 (e.g., frequency)
- An upper bound for MOD1 (e.g., frequency)

This reflects calibration instruments containing ranges of voltage and frequency. Interpolations are generally not valid between different ranges of voltage and possibly frequency. Interpolations and extrapolations are generally valid as long as ranges do not change.

Each range lists a number of data points that can be used for interpolation. Data points are sorted by MOD1 value. After each MOD1 value, one or more Nominal values and their corrections follow. For example:

```
RANGE: 3 19.9999  100  10e3
 MOD1: 100
       3  3.0002, 10 9.99998 , 19 18.9997
 MOD1: 1e3
       3  3.0002, 10 10.0001 , 19 18.9999
```

You may have a single Nominal data point for a given MOD1 value and there may not be an identical Nominal data point for another MOD1 value. In that case, the

NOMINAL and MOD1 values in the correction file must exactly match the values in the calibration procedure to determine the corrected value.

You may have a single Nominal data point for a given MOD1 value, and there may be an identical Nominal data point for another MOD1 value. In that case the Nominal data point value must exactly match the Nominal value in the calibration procedure in order to determine the corrected value.

You may only have data points for a single value of MOD1. In that case the MOD1 value in the correction file must exactly match the MOD1 value in the calibration procedure to determine the corrected value.

In all other cases, data points must be on a rectangular grid as a function of Nominal (voltage) and MOD1 (frequency) value. The data points must also be in sorted order: increasing values of nominal with increasing values of MOD1 values. (Refer to the example later in this chapter.)

The process of determining corrected values is as follows:

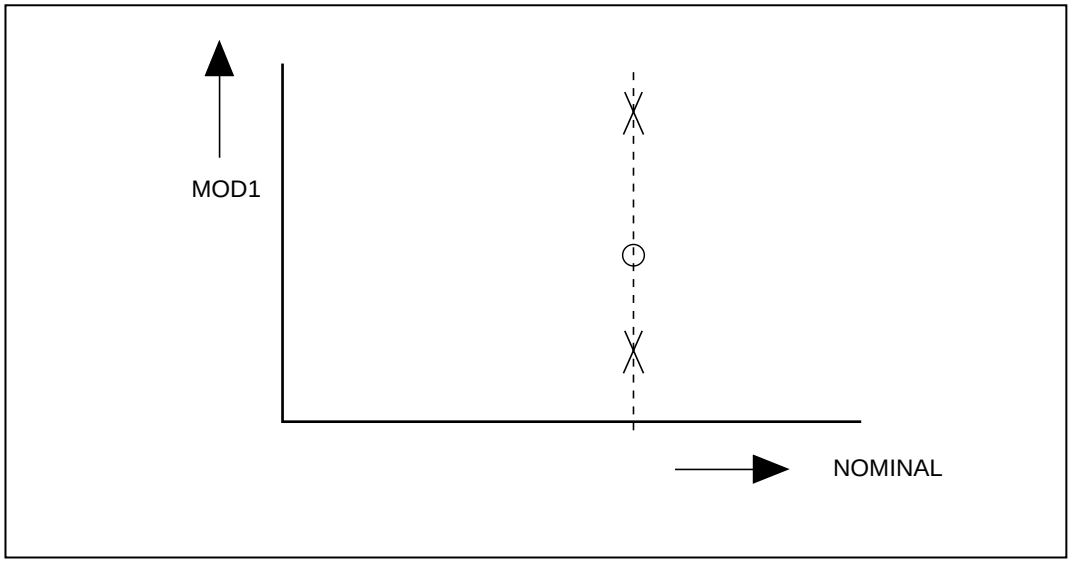
- Check to see if there is a range available over which interpolations can be done; if not, try the next range. Ranges may be overlapping. If all ranges fail, no correction is possible.

  The accuracy file should reflect standard instrument accuracy for ranges not covered by the correction file. There may be overlapping ranges of correction. The most desirable range to be used for interpolation should be shown first in the correction file.

- Check to see if there is a data point that exactly matches both the Nominal value (e.g., voltage) and MOD1 value (e.g., frequency). If found, return the correction; if not, continue the interpolation process.

- Check to see if there are at least two data points with the same Nominal value but different MOD1 values (e.g., frequency) in the list of corrected values in the range selected.

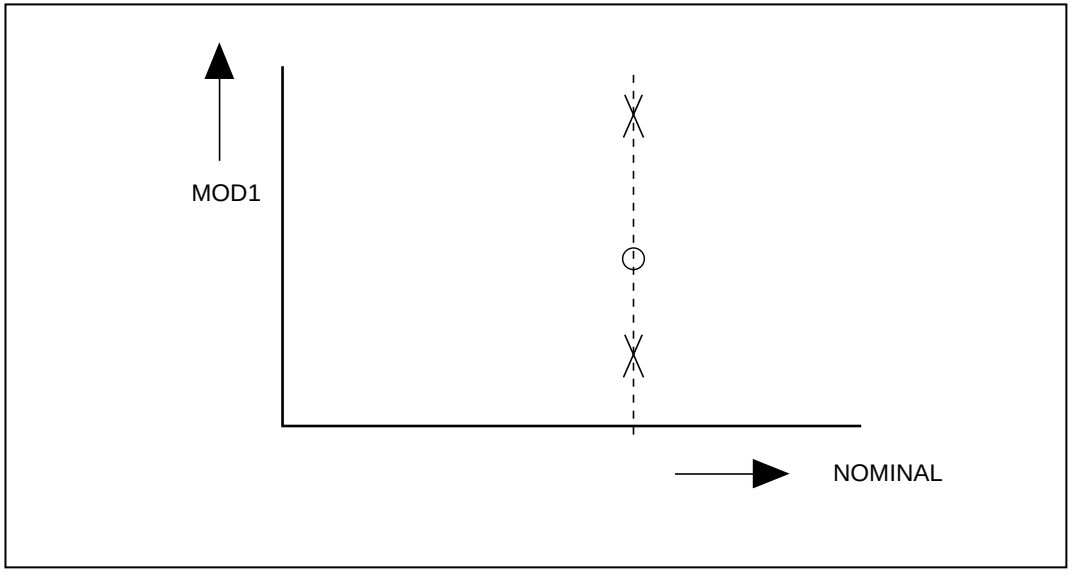
  1. Find the MOD1 values that are closest to the desired MOD1 value above and below the desired value of MOD1. See Figure 7-1.



IL701F.EPS

Figure 7-1. Selection of Data Points in a Range of the Correction File

If no MOD1 value below or above can be found, find the next MOD1 value above or below the desired value to be able to extrapolate. See the following figure.



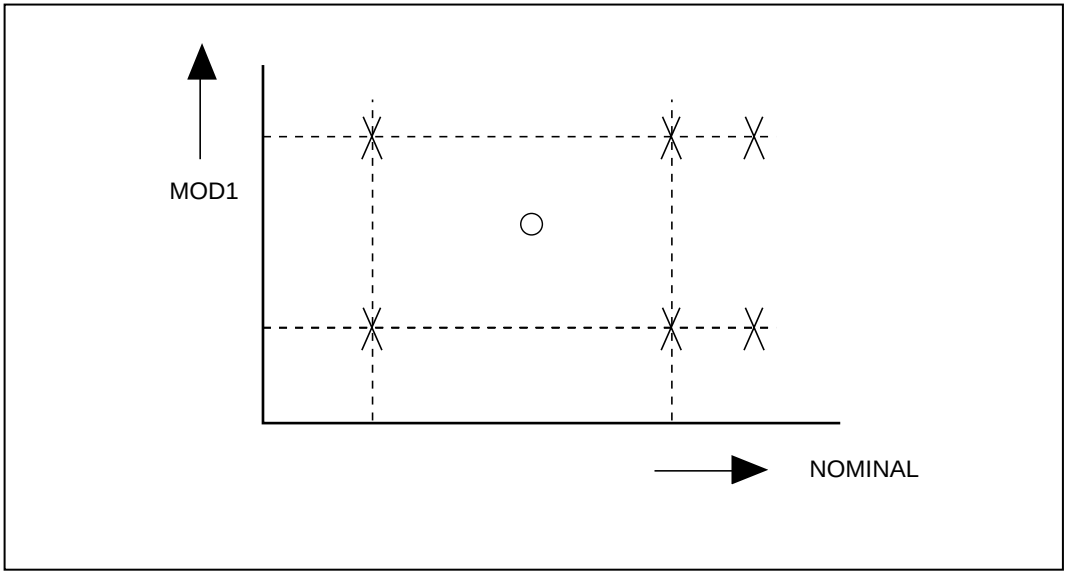
il702f.eps

Figure 7-2. Data Points and Desired Value Requiring Extrapolation

If extrapolating, find the difference between the closest Nominal value and the desired Nominal value, and the difference between the two reference Nominal values. If the ratio is more that 0.2, extrapolation is disallowed, and the correction within the range is not successful.

2. When suitable data points are found, compute by linear interpolation the corrected value of the nominal at the MOD1 value desired. If conditions are not satisfied, continue the interpolation process.

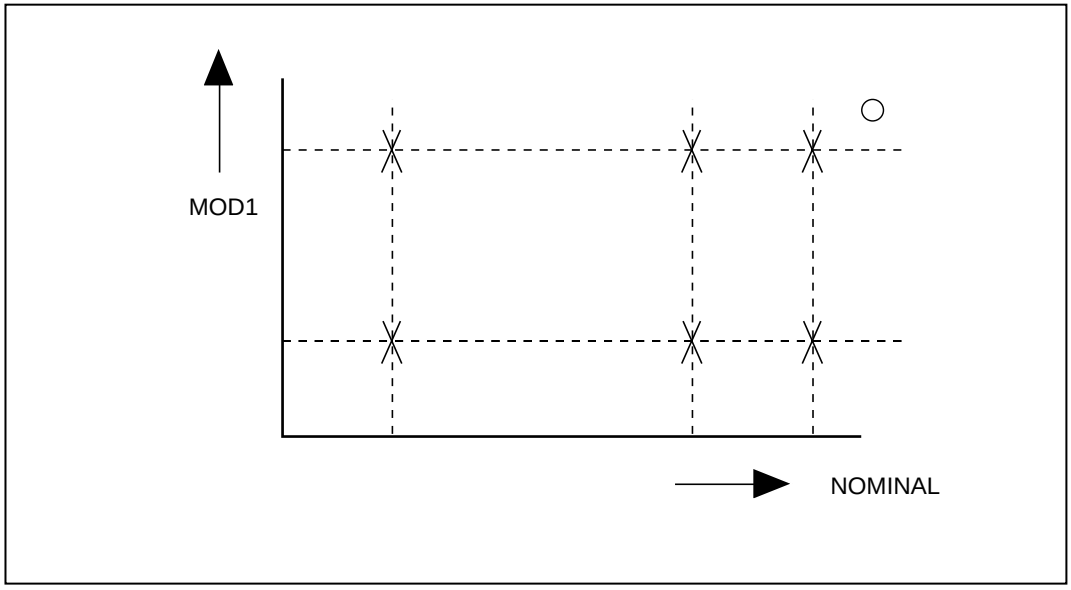
- Check to see if there are at least two data points with the same MOD1 value but different Nominal values (e.g., voltage), in the list of corrected values in the range selected. Find the Nominal values that are closest to the desired Nominal value. The data point selection is similar to the previous case where the Nominal value was fixed and where two values of MOD1 existed. If conditions are not satisfied, continue the interpolation process.
- Data points must be on a rectangular grid when the data points are plotted out as a function of Nominal (e.g., voltage) and MOD1 (e.g., frequency), unless the range contains just one or two data points. First, you search for two values of the MOD1 closest to the desired MOD1 value above and below the desired MOD1 value. Next, you search for the Nominal values above and below closest to the desired Nominal value. See the following figure.



il703f.eps

Figure 7-3. Data Point Selection

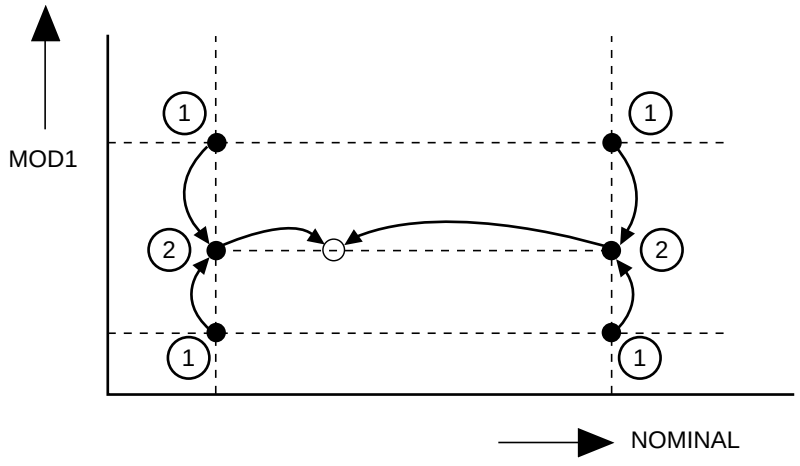
If you cannot find two values of Nominal and/or two values of MOD1 above or below, find the closest values of Nominal and/or MOD1 both above or below, and try to extrapolate. See the following figure.



il704f.eps

Figure 7-4. Data Point Selection and Extrapolation

The range of extrapolation is limited to 20% of the difference of Nominal or MOD1. If the desired value is not within that range, the extrapolation scheme fails, and no correction is applied. The interpolated correction value is computed as follows. See the following figure.



il705f.eps

Figure 7-5. Correction Computation Method

1. Find the correction factors for the desired Nominal values at the lower and higher value of MOD1 (four correction factors).
2. Compute the two interpolated correction values for the desired Nominal value at the lower and higher value of MOD1 (result is two new correction factors).
3. Compute the correction at the desired combination of Nominal and MOD1 is found by linear interpolation from the new correction factors.

Using Correction Files

Correction values are used in the results reporting and loading MEM or MEM1, depending on the type of test (Evaluation, Setup, Nominal Set, Compare) and mode (Slew, Keyboard Entry, Go/No-Go).

The effect on memory locations MEM and MEM1 and the UUT Indicated and System Actual values is as follows.

```
Evaluation Test:  MEM : no change
(MOD4 = blank)    MEM1: Slew    : last slewed value
                                  (CORRECTED)
                        Keyboard: entered value
                        Go/No-Go: Nominal value
                                  (CORRECTED)
                                  Test limits are adjusted
```

```
                UUT Indicated: Slew    : Nominal value or MEM
                                         (Not Ohms)
                               Slew    : last slew value (Ohms)
                               Keyboard: entered value
                               Go/No-Go: OPER-EVAL

                System Actual: Slew    : last slewed value (Not Ohms)
                                         (CORRECTED)
                               Slew    : Nominal value or MEM value
                                         (Ohms) (CORRECTED)
                               Keyboard: Nominal value
                                         (CORRECTED)
                               Go/No-Go: Nominal value
                                         (CORRECTED)

Setup Test:          MEM : no change
(MOD4 = S)           MEM1: Nominal value or MEM
                           (CORRECTED)
                           Calibrator output is not adjusted.

Nominal Set Test: MEM : Slew    : last slewed value
(MOD4 = N)                       (CORRECTED)
                  MEM1: no change

Comparison Test:  MEM : no change
(MOD4 = C)        MEM1: Slew    : last slewed value
                                  (CORRECTED)
                UUT Indicated: Slew    : MEM

                System Actual: Slew    : last slewed value
                                         (CORRECTED)
```

Rules for Correction Files

The following are the rules for correction files:

- Correction Files are ASCII files that can be modified using any text editor.
- Text is not case-sensitive.
- The correction file may contain any number of blank lines or comment lines.
- Comment lines have a pound sign (“#”) as the first non-white space character.
- White space before the beginning of text or after the text of a line is not significant.

- Each file is a sequence of sections. Each section consists of a MODE line, followed by one or more ranges.
- The MODE line has the following format:

  - `MODE: <unit specifier> [ <flag> <flag> .....]`

- The list of valid unit specifiers depends on the instrument. For example, the valid unit specifiers for the 5100 FSC are Volts and Amps.
- Flags are also specific to each instrument. For the same 5100 FSC, valid flags are E (for accuracy enhancement) and O (for Divider Override).
- A MODE line may not have extraneous characters.
- A range of values is defined by the following format:

```
Range: <nlb> <nub> <mub> <mub>
```

where:

nlb is the Nominal lower bound

nub is the Nominal upper bound

mlb is the modifier lower bound

mub is the modifier upper bound

Example:

```
Range: 2 19.99999 40 20
```

The string “NA” may be used to indicate that a particular value for the modifier is not applicable and is not used.

- The lower and upper bounds for Nominal and MOD1 must always be specified. They typically coincide with a range of a calibration instrument.
- Data points associated with a range follow the Range definition starting with a new line. Data points are listed by increasing MOD1 (frequency) value. The MOD1 value for which data points are valid is shown by:

  ```
  MOD1: <mvalue1>
  ```

  Starting on the next line, Nominal data points and their corrections are shown. A comma “,” separates each group of Nominal value and corrected value:

  ```
  <nvalue1> <cvalue1> , <nvalue2> <nvalue2> , ...
  ```

A range and data points may be shown as:

```
Range: 2 19.9999 40 2e3
 Mod1: 100
  3 3.0002 , 10 9.9998 , 19 18.9997
 Mod2: 1e3
  3  3.0002, 10 9.9999 , 19 19.00003
```

Example Correction File for 5200A

```
# Example Fluke 5200A Correction File (partial)
# 5200.COR
#

MODE: VOLTS

Range: 50e-3 119.9999e-3 50 100e3
 MOD1: 50
    50e-3 49.9998e-3, 100e-3 99.9976e-3
 MOD1: 100
    50e-3 50.0000e-3, 100e-3 100.0013e-3
 MOD1: 200
    50e-3 50.0002e-3, 100e-3 100.0023e-3
 MOD1: 1e3
    50e-3 50.0022e-3, 100e-3 100.0023e-3
 MOD1: 2e3
    50e-3 50.0023e-3, 100e-3 100.0030e-3
 MOD1: 10e3
    50e-3 50.0026e-3, 100e-3 100.0057e-3
 MOD1: 20e3
    50e-3 50.0033e-3, 100e-3 100.0033e-3
 MOD1: 50e3
    50e-3 50.0040e-3, 100e-3 100.0112e-3
 MOD1: 100e3
    50e-3 50.0020e-3, 100e-3 100.0263e-3

Range: 500e-3 1.99999 50 100e3
 MOD1: 50
    0.5 0.499998,  1.0 0.999976

Range: 3 11.99999 50 100e3
 MOD1: 50
    3.0 2.99989,  10 9.99970

Range: 30 119.9999 50 100e3
 MOD1: 50
```

```
    30 29.9997,   100 99.9992

Range: 300 1199.999 50 50e3
 MOD1: 50
    300 299.995,  1000 999.980
```

Chapter 8
Configuring the System

Contents **Page**

Introduction

The usage model in Chapter 1 describes Fluke Metrology software as having the following user categories: — procedure writer, operator/data entry, manager, and administrator. This chapter is written specifically to assist the person who is to perform general administrative tasks using the tool provided for these tasks, the DB Setup application.

Keep in mind that all of the administrative tasks are not concentrated in this application. For example, backing up the database on a regular basis (and restoring it if necessary) are system administration tasks that are performed using tools provided in the Sybase package. Backing up the database is described in Chapter 2.

The information in the rest of this introductory section is needed by anyone using the database, such as connecting to the data source. Most, however, are tasks only the administrator should ever do, such as adding a user or setting up the date format. The tasks normally performed by the other users are covered in other chapters, but are repeated here for the convenience of the user who performs the administrator functions. The administrator should strive to become proficient at all of these in order to train others and to answer their questions.

Starting an Application

Each time you want to access the metrology database for any reason, you must first connect to it as a data source. This is required by ODBC. The database engine itself arbitrates and controls access to the database (as opposed to these controls being externally applied by every application desiring to view or modify the data).

To connect to the database, proceed as follows:

1. Make certain that the Sybase SQL server is running.
2. If you are on a single workstation system (not networked), you can simply observe if the Server icon is minimized.

   If you are running on a network server, check to see if the server hardware is up. If it is, the server software is probably running, too.
3. On the client workstation, start up the data requester. The icon appears.
4. When Sybase SQL locates the data source, it displays a dialog box asking for the user ID and password. Enter the name and password and press **ENTER**.
5. Start the client application you will be using by clicking its icon.

The Database Setup Application

Most system-level configuration is done using the Database Setup application (for brevity, often called just “DB Setup”). The most prominent example is the

customization of fields and forms, covered later in this chapter. First, however, this chapter explains the managerial or administrative functions provided by DB Setup.

Changing the MT Password

Every copy of MET/BASE is shipped with the same Administrator password. When the software arrives at your site, the administrator's User Name and Password are both **MT**. The administrator's User Name is set, but for security reasons, the password should be changed immediately.

Change the MT password as soon as possible after you are confident that you can reliably connect to a data source.

The MT user has unequaled authority within the MET/BASE system. This responsibility should be treated with great care.

Caution

Change the MT password as soon as possible after verifying that you can reliably log on. The MT user has full authority within the database, so the MT password should be treated with care.

To change the password of the MT user, proceed as follows:

1. Make certain that the database engine is running.
2. On a client workstation, start the Data Requester.
3. Start DB Setup.
4. Log in to your data source with the user ID **MT** and the password **MT**.
5. When the DB Setup application starts, open the **Security** menu and choose **Change Password**. You must type the new password twice, to guard against typographical errors.

WARNING

DO NOT FORGET THE PASSWORD FOR THE MT USER. WITHOUT THIS PASSWORD, IT IS IMPOSSIBLE TO GAIN ACCESS TO THE CONFIGURATION APPLICATION OR TO ISQL COMMANDS THAT ARE REQUIRED FOR MANAGING THE DATABASE.

FLUKE CANNOT RECOVER ANY DATA YOU MIGHT LOSE DUE TO A FORGOTTEN PASSWORD. THIS HOLDS FOR ALL USERS, INCLUDING THE MT USER.

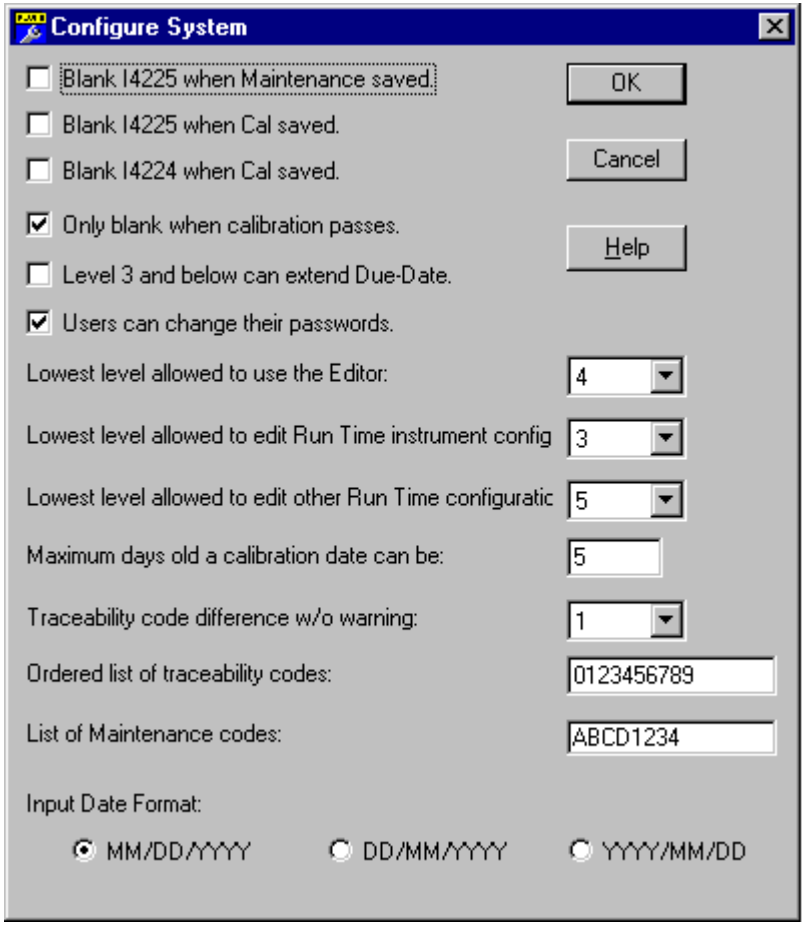
Setting the System Configuration Options

As a rule, the system configuration options only need to be set up one time because the settings are not likely to change from day to day, but are more permanent in nature.

Make sure the Server is running before you begin.

1. If necessary, start the Data Requester, then open DB Setup.
2. Log in as MT, and provide the MT password. Only the MT user is permitted to use the DB Setup Application.
3. When DB Setup opens, go to the Configure menu and select **System.**

A dialog box appears, as shown in the following illustration.



zv801s.tif

Figure 8-1. System Configuration Options Dialog Box

The following paragraphs explain the System Configuration options.

Blank I4225 when Maintenance saved
Blank I4225 when Cal saved
Blank I4224 when Cal saved

Field 4225 (Status) of the Inventory table is intended to store the temporary status of an asset. In larger installations, this field can be used to indicate that the asset has arrived for calibration, is on a bench, is undergoing testing, or other process milestones.

Field 4224 (Arrive Date) is intended to store the date the asset arrived.

If assets move through your process rather quickly, you may wish to simply disregard either or both of these fields. To do so, configure the field to be blanked. Now the status or date is blanked each time a new calibration or Maintenance event is recorded in the database. By checking these boxes a null value is stored in the field, signifying that you do not store temporary status or arrival date information.

On the other hand, if you do not blank I4225, you can use the eight characters available to indicate the temporary status of your assets; by not blanking I4224, you can use the date field to store the date of arrival of the asset. In this case, any query or report that checks the value of these fields can give an up-to-the-minute indication of the whereabouts of the asset in your process.

Level 3 and below can extend Due-Date

Users having access level 3 and below are not usually permitted to extend the Due Date past the date calculated by the software. Levels 4 and 5 can always extend the Due Date. Checking this box permits levels 3 and below to enter dates after the calculated due date. The Administrator should make this determination based on the established practices at your site.

Note

Customization can restrict which lower level users can access this field.

Users can change their passwords

When selected, all users are able to change their own passwords. If left blank, only the MT user can change passwords. Your choice should probably be influenced by such factors as number of users and personnel turnover rate.

Lowest level that can edit procedures

The Calibration Procedures you develop and maintain in your lab are directly responsible for proving the traceability of your instruments' calibration histories. This option permits you to restrict access to the ability to edit them.

Lowest level that can edit validation

As with editing calibration procedures, editing validation presents a direct influence on your lab's credibility. This option permits you to restrict access to the ability to edit the validation files.

Maximum days old a calibration date can be

Every time a calibration date is entered, the software checks to be sure it is not too old. Each calibration date is checked against the system date. Operators with access levels 1, 2, and 3 can only enter cal dates of today's date minus the number of days you set here. For example, the default is seven days; if today is the 18th, these operators can not enter a date earlier than the 11th. To relax this check, enter a larger number. A setting of 30 days may be more realistic in some situations. This would let all operators input any calibration date up to one month old.

Traceability code difference without warning

When you enter a standard used to calibrate an asset, the software checks to be sure that the standard has a sufficiently high traceability code. You assign these codes to each asset to reflect the relative accuracy of classes of standards. The specified difference refers to the difference in traceability codes of the asset and any standard. The Fluke Metrology software does not enforce the difference, but it does issue a warning if it is greater than specified here.

The traceability code difference is not enforced because a standard for one procedure may not be considered as a standard for other procedures. The difference is simply a check to warn of possible traceability violations.

The difference can range between 15 and -15. Normally, a difference of 1 would be an appropriate and useful setting. This would mean that a warning would occur if a standard's code was not at least one code higher than the asset being calibrated. If your standards are fairly well defined and you want a strong check, you could choose a larger difference, say 2 or 3.

A very weak check can be achieved by setting the difference to 0 or -1. Zero means that a standard with the same code as the calibrated asset could be used, and negative one (-1) indicates that an asset could be used to calibrate a standard one level higher than itself.

To completely disable the warning, enter -15 (negative 15). This setting allows any asset to be entered as a standard without a warning.

Ordered list of traceability codes

Traceability codes are an ordered sequence of single characters that define the traceability hierarchy for your asset inventory. They represent classes of standards, and are therefore loosely related to accuracy.

The codes come into play when you list standards for a calibration record. If the difference between the positions of the codes in the ordered sequence is less than a pre-set value, the software displays a warning.

In this selection, you define the traceability codes that the software will use for your database. You can specify up to 15 codes, with the order descending from left to right. For example, the letters RCTWPN could be your ordered list with the letters carrying the following meaning:

R Reference standard
C Consensus standard
T Transfer standard
W Working standard
P Production standard
N No calibration required

Or you could just leave the default sequence 12345, or invent some other scheme that you'll be able to remember easily.

Maintenance Type Codes

As with Traceability Codes, Maintenance Type Codes are open to your own definition. You can establish up to 32 different maintenance codes.

This field is used in situations when it is desired to generate "Due for Maintenance" reports. Being able to specify as many as 32 different codes lets your individual site establish a system of preventive maintenance or other periodic maintenance tracking unique to your own requirements.

The database covers ALL the assets, even those that are not directly involved in calibration, like bar code readers or CD recorders or players. The Maintenance type codes are validated at the "RT" field (3251) on the Maintenance results form, permitting you to generate reports having meaning specific to the asset or asset type.

For example, you might use the numbers 1 through 32, assigning a particular meaning to each code for all your laser printers and an entirely different meaning for the bar code readers. Since asset Maintenance history records also include an interval specification field, each of 32 different types of events can be scheduled for each asset. There is one active due date for each Maintenance code.

Date format

In the US, the most commonly used date format is MM/DD/YYYY, while in Europe the standard is DD/MM/YYYY. Other areas use YYYY/MM/DD. The buttons at the bottom of the System Configuration dialog box permit selecting any of these formats. The format can be toggled on the fly. Ordinarily you set it once and leave it set for the normal format used at your location.

Maintaining User Information

Maintaining the user information is an ongoing task. While you may not often add a new user, still you should give some thought to the process. Your main concerns should be to establish and maintain an appropriate access level for each user, and assign each user to the correct segregation dataset if you have structured your data in that way.

The following table describes the permissions that are set up for each of the user levels.

Table 8-1. User Access Levels

Access Level	Authorized Actions
MT	No permissions for adding or changing data; but MT is the only user authorized to run the DB Setup application. The MT user can delete or modify the entire database.
5, 4	These two levels perform supervisory and error correction functions, and write Calibration procedures. They have read and write permission for the entire database, including historical records. Range checks are not made for most data fields. The software does not differentiate between levels 4 and 5, so you can use these numbers as you wish.
3, 2, 1	Basic data entry/technician. Persons holding these access levels can add new Inventory and Calibration records, and in MET/TRACK applications, can add Maintenance, and Location records. Level 1, 2, and 3 users cannot alter historical data. These levels are appropriate for most users. The actions these three levels can perform are established through customization.
0	Read only. Persons with level 0 access can look at data but have no write permission.

Caution

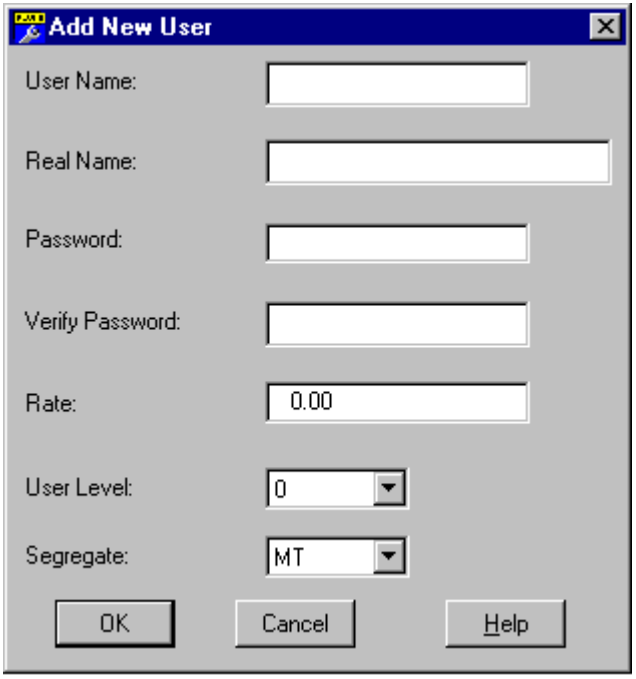
Don't lose or forget your user name and password! Relay this caution to all users, impressing upon them the importance of remembering passwords to ensure continued access to the database.

Choose every user name and password carefully. Fluke cannot recover passwords. Select a password that you can remember. Remember that user names and passwords are case sensitive.

Adding Users

1. To add a user, first be certain that the server is running, then complete the following steps:
2. Start the DB Setup application, and connect to your database as the MT user.
3. Under the Configure Menu, select **Configure**, then **Users.**

4. A dialog box displays that looks like the one in Figure 8-2.



zv02s.tif

Figure 8-2. Add User Dialog Box

5. Fill in a user name and then enter the full name (Real Name) of the new user. In the database, the person's real name is a permanent record to identify the user. This must be a unique name.

   An operator may have multiple user ID/password level combinations, for interacting with the software at different levels. Every user name must be unique, although passwords could be the same.

Note

Since a user can have various combinations of user name and password, you might consider adding the level number as a suffix to the user name. For example, MARIA3 and MARIA5 could be the names Maria uses for logging in at levels 3 and 5 respectively. A very good reason to also assign level 3 for level 4 and 5 users is that at level 3, all of the software's data checking is active. Levels 4 and 5 are assumed to be more aware and ***wary*** *of actions that might cause problems.*

6. Enter a password for the person to use for the first login. Passwords must be at least two characters long. You must type the password twice as insurance against making a typographical error.

Caution

To ensure security, you must remember your password! Guard passwords carefully. Do not share or write them down anywhere that they are accessible to others.

Fluke CANNOT recover your password if you forget or lose it. Password creation is a one-way process. Instruct all users to change their password the first time they log in, and to be sure to remember the password.

7. Next select the access level using the pull-down selection list.
8. Finally, if you are using the database segregation feature, assign the operator to the appropriate segregate. If you have not yet decided about segregation, you can safely ignore this selection for the time being. However, if you later decide to implement segregation, you'll have to revisit each user with the **Modify User** menu selection to set up their assignments.

   For more information about segregation, see the heading "Data Segregation" further on in this chapter.

Adding Yourself as a "Normal" User of the System

In this meaning, "normal" users are those who add, change, and view the data. It is important for the administrator to have access to the data as well as to the configuration, so you should add yourself as a level 5 user as soon as possible after verifying that you can reliably log into the various applications. In this way, you will be able to run all of the applications, making you much more efficient at checking changes you make as MT.

Modifying a User

From time to time it might be necessary to change a user's settings For example, when an employee is promoted, that user might require a higher access level, or you may want to move someone to a different segregation group. Remember that you can change the login name, but the user's **real name** is a permanent record and cannot be changed.

To accomplish a name change, demote the original name to access level 0 (and assign the STD segregate if you are using the segregation feature), then add a new user under the new name. Names and user names are never deleted from the database.

To modify user settings, first be sure that the Server is running, then proceed as follows:

1. Start the DB Setup application , and log into the database as user MT.
2. Under the **Configure** menu, click and hold **Users**, then choose **Modify Users.** You will see a list of all the current user names in the system.
3. Highlight the name you wish to modify and click **Ok**. A dialog box appears much like the one that you see when you add a new user, containing the current information for the user you selected.
4. Edit the fields you want to change, then click **Ok** to save the changes and exit the dialog.

Blocking User Access

If a user has entered data, they are never deleted from the database. If you wish to block an individual's access, you have two options, as follows:

1. Assign the user an access level of zero. With this option, the user retains read permission on the database.
2. Change the password without giving the new password to the user. This prevents that user name from ever logging in to any of the Fluke Metrology applications.

Configure Company

When you installed MET/BASE, you were asked to supply your company name. This entry is specific to each work station and is used for the header of any calibration procedures that you create.

The Fluke Metrology software provides another location for your company name for substituting into reports. When you create a report in *Crystal*, you can automatically substitute the contents of these fields into your reports. This lets you create reports that can be run on other databases, confident that the correct company name will appear at the top, but without expending extra effort to modify the report.

Changing the Contents of the Configuration Table

From the DB Setup application, select **Configure,** then **Company.** A dialog box opens, showing you the current contents of the DBCONFIG table. This table is pre-loaded with entries that generally describe the intended final output. For example, as received, the database contains the text "Your Company Name Goes Here" in the field named MT_Company.

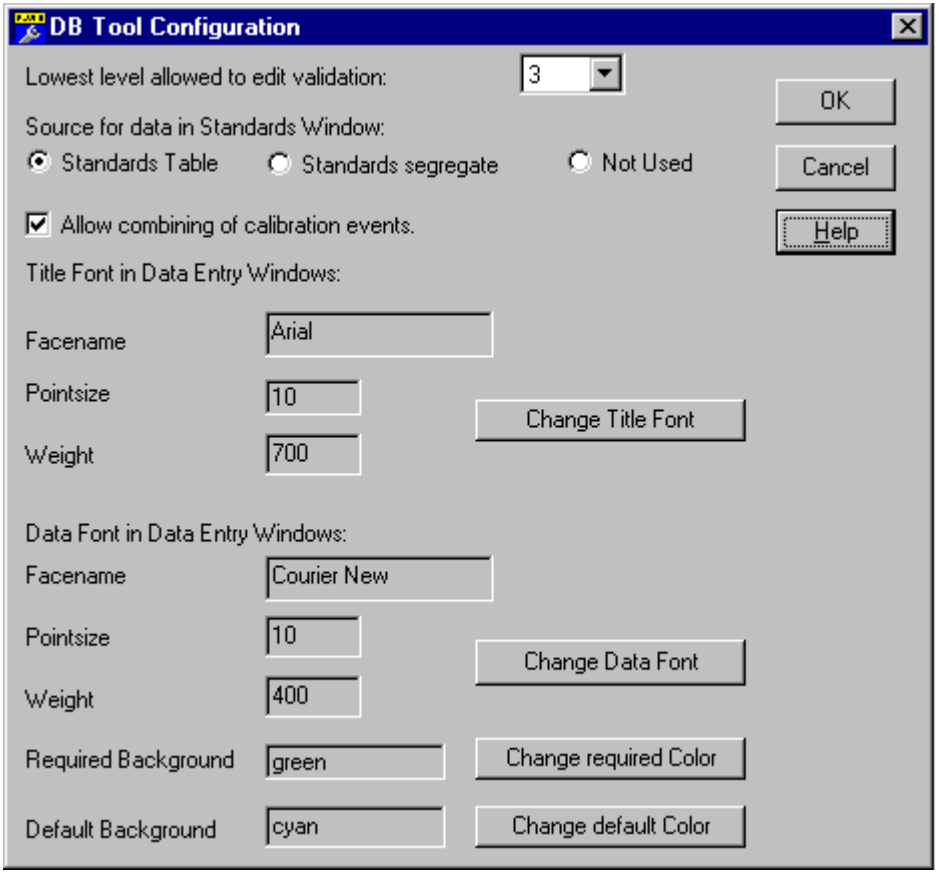
The DBCONFIG table contains 5 other "user" fields, called MT_User1, MT_User2, MT_User3, MT_User4, and MT_User5.

Using DBCONFIG Data in a Report

You can use the MT_User<n> fields for any purpose, keeping in mind that their contents are intended for eventual substitution into a report. For this reason, any entries you make should use regular alphanumeric text characters. If you use quote marks, or the apostrophe character, the substitution will fail.

Data Forms

DB Setup allows you to configure data forms as shown in the following dialog box.



zv810s.bmp

Special Features

Non-Due Days

MET/CAL allows you to exclude certain days from the calibration due-date pool. This way, an asset never comes due for calibration on a holiday or other non-working day. By opening DB Setup and selecting **Configure**, then **Non-DueDays**, you can specify days where calibration due dates will never occur.

Uncertainty

MET/CAL allows you to specify the uncertainty configuration for the entire site. Select **Configure**, then **Measurement Uncertainty**. These parameters can be overridden at the workstation level by setting them in the metcal.ini file, or at the procedure level by setting them in the procedures. See the README file on the MET/CAL CD for more information.

Coverage Factor

The coverage factor is the value multiplied by the standard uncertainty to get the expanded uncertainty.

Number of Measurements

You can set the number of measurements that are taken in a calibration procedure. Enter any number, other than a zero, to enable measurement uncertainty.

Note that you can set the number of measurements to one. However, with one measurement, the standard deviation of the readings cannot be computed. This means the metrologist must have some prior information about the test being performed.

If you enter a zero as number of measurements, you disable measurement uncertainty and the run time works as it did in prior versions. See the README file on the MET/CAL CD for more information.

Use Student's T

This can be set to either Y to enable it or N to disable it. If set to Y, then the standard deviation of the measurements are multiplied by a factor relating to the degrees of freedom in the test. See the README file on the MET/CAL CD for more information.

Data Segregation

The segregate feature allows you to divide your database into separate, mini-databases for some users while keeping all of the data together. This ensures traceability, ease of database backup, and system management.

For example, you might divide the database into electrical and mechanical segregates. Then, electrical users could access the data for electrical assets, but could not access any mechanical assets. Mechanical users could access the data for any mechanical asset, but not electrical assets. You, as the calibration lab, could see and work on all of the assets in the database.

You might also segregate data based on your customers. You assign each customer site a segregate value. This allows customers to log into the system and view the status of their assets, but prevents them from seeing what other customers have.

Set up segregates using the DB Setup application. Select configure segregates from the main menu and add a segregate name. Once you have added segregate values, you can associate users, assets, and customers to a single segregate.

The add user screen, add asset screen, and add customer screen all have a segregate field that you can assign to one of the values you created in DB Setup. Note that it is possible for a user to add an item to a segregate other than their own, but they will never be able to find the item again. Avoid this by using the meta value U_SEG in the segregate field, which forces the user's segregate into the field.

You can also assign reports to a segregate. This means that, if you have specified a segregate for the report, only user in that segregate can execute that report.

You can assign reports to the <any> segregate, which allows any user to run the report. A report that is written using table linking must be created for a specific segregate. So, in addition to assigning it to a segregate in the add reports dialog, you must write the report for that segregate in Crystal Reports.

Reports written to run off stored procedure output can be assigned to the <any> segregate, or to a specific segregate. Because the report works off a stored procedure, any user can execute it.

There are two special segregates. They are MT and STD. The MT segregate contains all of the data in the database. Any user who is assigned to the MT segregate can access all of the data. Typically, users in the cal lab are assigned to the MT segregate.

The STD is a special segregate intended to be used for standards. Since one segregate can not "see" another segregate, this means that you would have to duplicate all of the standards in each segregate. You can overcome this by putting standards in the STD segregate. Assets in the STD segregate can be used by other segregates as standards in calibration events.

As Found/As Left

As Found/As Left represents a calibration philosophy about how calibration is performed. MET/BASE supports three basic philosophies. Which philosophy you choose determines how the Found/Left field (2308) is used.

Philosophy 1

When an instrument comes in for calibration, you run the verification procedure. If it passes, you record the data and return the instrument. If it fails, you may or may not record the fail data, calibrate/fix the instrument, and run the verification procedure again. (It passes this time - you just fixed it).

Following this approach, you would leave Field 2308 blank (or suppress it), and the user doesn't care about As Found/As Left.

Philosophy 2

You ALWAYS record As Found/As Left data. When an instrument comes in for calibration, you run the verification procedure. Whether it passes or fails, you record the data for the instrument with 2308 set to AS-FOUND, since this is how the instrument came into the lab. If it failed, you calibrate/fix the instrument. Now, if it passes, you run the verification procedure again, but this time you record the data AS-LEFT, since you are sending the instrument back to the user.

Philosophy 3

You record As Found/As Left data, but don't run the verification procedure twice if the AS-FOUND passes. This approach is the same as #2 if the instrument fails on the AS-FOUND. If the instrument passes on the AS-FOUND, then 2308 is set to FOUND-LEFT, to indicate that the instrument was FOUND in this condition, and it LEFT the lab in the same condition.

The difference between #2 and #3 may not seem very large, but it becomes important when you want to report on the data, and you are trying to match up As-Found and As-Left data in Crystal Reports.

The purpose of field 2308, Found/Left, is to permit you to tag each calibration event with information about when the event occurred in your process. The tags specify whether the results indicate the condition of the instrument at the time it arrived, when it left, or both. Additionally, you can disable the feature so that no such tagging takes place.

However, note that by not using the Found/Left feature, you also are not able to generate a reliability coefficient for your instruments. This is a figure of merit that indicates each instrument's reliability, and is stored in field 2304. Because of the connection between fields 2308 and 2304, these fields make up a "Calculation Field Group"; the database enforces the interaction between these two fields.

Constructing a reliability coefficient is the next topic, so if you are unsure whether to implement Found/Left, you might skip ahead to the next heading for details about the Reliability field.

The Found/Left field (C2308) stores a single value for each calibration event, and the feature works the same way whether the data comes from the Run Time or the Manual Entry application.

The Found/Left field is 10 characters, and it must contain one of the values **AS FOUND**, **AS LEFT**, **FOUND_LEFT**, or it must be empty (contain a null value).

The purpose of field 2308, Found/Left, is to permit you to mark each calibration event with a tag specifying the point in your process that the event occurred. The tags specify whether the calibration results indicate the condition of the instrument at the time it arrived, at the time it left, or both.

The As Found / As Left feature makes use of prompts contained in both the FAIL_DB.TXT and PASS_DB.TXT files (they are in the main directory — typically C:\METCAL). These prompts read as follows:

```
As Found / As Left  :: FOUND_LEFT$ 2308
```

The effect of these prompts is to create a validated field on the Pass and Fail Post Run prompt screens (assuming that the validation entries for 2308 have not been modified). The choices presented are the legal values, including the null value. If you should modify the validation of 2308, the automatic loading of the Reliability field will not operate.

To disable the feature, use a text editor such as Notepad, or the Procedure Editor application to "comment out" the prompts in both the FAIL_DB.TXT and PASS_DB.TXT files in the main (METCAL) directory. A pound sign (#) at the beginning of the line causes the Run Time application to ignore the entire line, thus removing the prompt from the post-run screen. If you decide to disable the feature, be sure to remove the prompts from both the FAIL- and PASS_DB.TXT files. In this case, you could also edit the validation entries for field 2308.

Constructing a Reliability Coefficient

If you have decided to make use of the Found / Left feature, you also have a way to affect the contents of the Reliability field, C2362, because the reliability field is a calculated field that receives information from two sources: Found/Left and Pass/Fail.

Providing As Found reports may be a requirement for some of your customers, but, even this is not the case, gathering reliability information may be a convenient way to alter calibration cycles for certain instruments. If an instrument with a 30-day calibration interval is always in cal upon arrival, you might feasibly extend the interval. Conversely, if you have critical instruments that usually arrive out of calibration, shortening the interval may be in order.

The following table shows the effect of As Found/As Left on the reliability field.

Table 8-2. Effect of As Found / As Left on Reliability Field

Pass / Fail (C2323)	Found / Left (C2308)	Reliability (C2362)
Pass ***or*** Fail	Don't Care	No Change
Pass	Found_Left ***or*** As Found	If rel < 0, set to ***1*** otherwise set ***rel=rel +1***
Fail	Found_Left ***or*** As Found	If rel > 0, set to ***-1*** otherwise set ***rel = rel -1***
Pass	As Left	No Change
Fail	Don't Care	No Change

Interval Types

As you enter assets, you can assign an interval type for each asset. The interval type for each asset is stored on the inventory form and is copied to the calibration form when calibration data is saved for that asset. The interval type can be changed at any time without affecting calibration history.

The interval type must be one of the following:

Type	Units	Interval
d	Days	24 hours
w	Weeks	7 days
m	Months	Calendar months
u	Uses	Number of uses
n	None	Never needs calibration

The interval type is important if you allow the system to calculate a due date rather than manually entering one.

For TIME interval types (Days, Weeks, Months), the system computes the due date using the cal date and the cal interval. If zero is entered, no due date is calculated.

For USE interval types, the system computes the due date when the number of uses (Field 4230) exceeds or equals the interval. The date this occurs becomes the due date.

For NEVER interval types, the interval is always zero. Use N for assets that never need calibration. A due date may still be entered manually, but, if the interval type is N, the software ignores the due date in the traceability reports. For purposes of forward traceability, an N interval type represents the termination of the traceability tree.

Abeyance

Abeyance is the condition of being temporarily set aside or in suspension.

MET/CAL provides a way to put a mechanical asset into abeyance when the asset is calibrated, but not used right away. Instead, it is placed in a special location where it stays until it is used. While it is in this location, it is in a state of

suspension or abeyance. When it is removed from the abeyance location, the due-date is then calculated.

There are six fields for abeyance, four for data retention. These fields are:

4258 : Abeyance state.

4259: Abeyance interval.

2380: Abeyance date.

2381: Abeyance time.

2382: Abeyance state.

2383: Abeyance interval.

Field 4258, Abeyance state, can be either Y or N. This flag determines how the due-date is calculated during calibration. If the flag is set to Y, then the asset will be placed in abeyance after it is calibrated.

Field 4259 is the abeyance interval. This is an interval to be used to set a due-date

in case the asset never comes out of abeyance. Fields 2382 and 2383 are copies of 4258 and 4259 and are stored for historical purposes when the calibration record is saved. Fields 2382 and 2383 are the date the item was taken out of Abeyance.

This is best explained with an example. Assume we have a device X. Its calibration interval is three months, and the abeyance interval (4259) is set to 24 months. Also assume field 4259 is set to Y.

We calibrate asset X on 1999-06-01. Field 2382 will be set to Y, and field 2383 will be set to 24. The due-date will be computed as 2001-06-01 (24 months out). Now, some time later (say 1999-08-01) we take the item out of abeyance.

To do this, we go to the inventory screen and set field 4258 to N. When we save inventory, the due-date for asset X will be recomputed to be 1999-11-01, field 2380 will be set to 1999-08-01 and field 2381 will be set to the current time.

Customization

Fluke Metrology software was conceived and designed by metrology professionals, people who know how a cal lab operates, and who understand the need for precision and accuracy in instrumentation and in record keeping. That kind of precision and accuracy is not possible if the software you use is a one-size-fits-all solution.

By knowing how a cal lab works, Fluke has been able to create software that integrates into your normal work flow, while providing flexibility, extensibility, and customization features. The goal is to provide you with a tool set that you can actually participate in designing. The final product is a custom solution without the overhead you would incur by building it from the ground up yourself, because it combines our best ideas and experience with your intimate knowledge of your processes.

The remainder of this chapter describes how to go about customizing the system so that it meshes perfectly with your organization.

As noted throughout this manual, the Fluke applications are designed to be useful right from the start. As soon as you install the software, you can begin recording your data and generating calibration certificates or reports. However, no two calibration laboratories have exactly the same requirements, and no two sites have quite the same vision about integrating the metrology lab with other processes or information sources.

Customization Defined

You have a great deal of control over the appearance of forms, and over the types of information that they gather. Changing field titles is only the beginning — you can change the location of fields on forms, and even specify other fields to receive the data. You can expand your database by adding tables of your own and link them into existing fields. Field validation lets you restrict input to a set of selections that you control, rather than chancing typographical errors or inconsistent input.

The Fluke Metrology software makes customization a topic you can address on your own schedule. You can do as much or as little customization as you want, and you can change your customization as the need arises and your requirements change. One of the many sample reports provided with the product allow you to view your current customization (filename MTCUST6.RPT).

In addition, the entire customization process has been made as simple to implement as possible. Using the familiar Windows user interface, all you have to do is point at the fields you want to change, and change them.

Caution

Without a good understanding of the reasoning behind the system as it exists prior to customization, it is possible to make a mess. It is possible to customize the system in undesirable ways, and even in ways that can actually cause you to lose data. If you have not familiarized yourself with the information in Chapter 1, be sure to do so before you begin your customization.

Customization addresses four issues, listed here in the order of their importance:

- The appearance and contents of forms
- The properties of fields
- The tab-order on a form
- Field validation

Field Types

Every field can be customized in any way that you wish. However, before doing ANY field customization, consider the following:

Each field is classified as CORE, CALCULATED, or AUXILIARY.

- **CORE** fields must be present, and must contain certain types of values for the underlying database to work properly. Examples are fields like 4201 (Asset Number) and 2301 (Calibration Date). Many of these fields can be set up to have default values. Even though the operator may not see them, they still exist and are used by the system.
- **CALCULATED** fields are those that have some sort of relationship to other fields. For example, 2303 (Due Date) is computed using values from 2301, 2302, and 2333 (Calibration Date, Calibration Interval, and Calibration Pass/Fail). There is no way to change the basic way these fields interact.
- **AUXILIARY** fields are all those others that are neither core fields nor are they part of a calculated field group; they just contain information. Fluke-supplied reports may expect certain types of information, but users can change these fields and modify the reports to their own purpose. Auxiliary fields are less sensitive to customization, but changes can still result in confusing reports or screens if you do not take into consideration the way the field is intended to be used.

Note

Most of this chapter explains operations that are performed using the Database Setup application. To avoid repetition, this chapter does not repeat the instructions to start the server if it is not running.

Core Fields: The Minimal Field Set

As previously stated, Fluke Metrology applications can be used just as they are shipped, without any special customization at all. This is so because initial values and defaults have already been established for all of the core fields. These fields represent the absolute minimum requirements to enable the various programs to interact with the database. Following is an illustration of the Calibration Form.

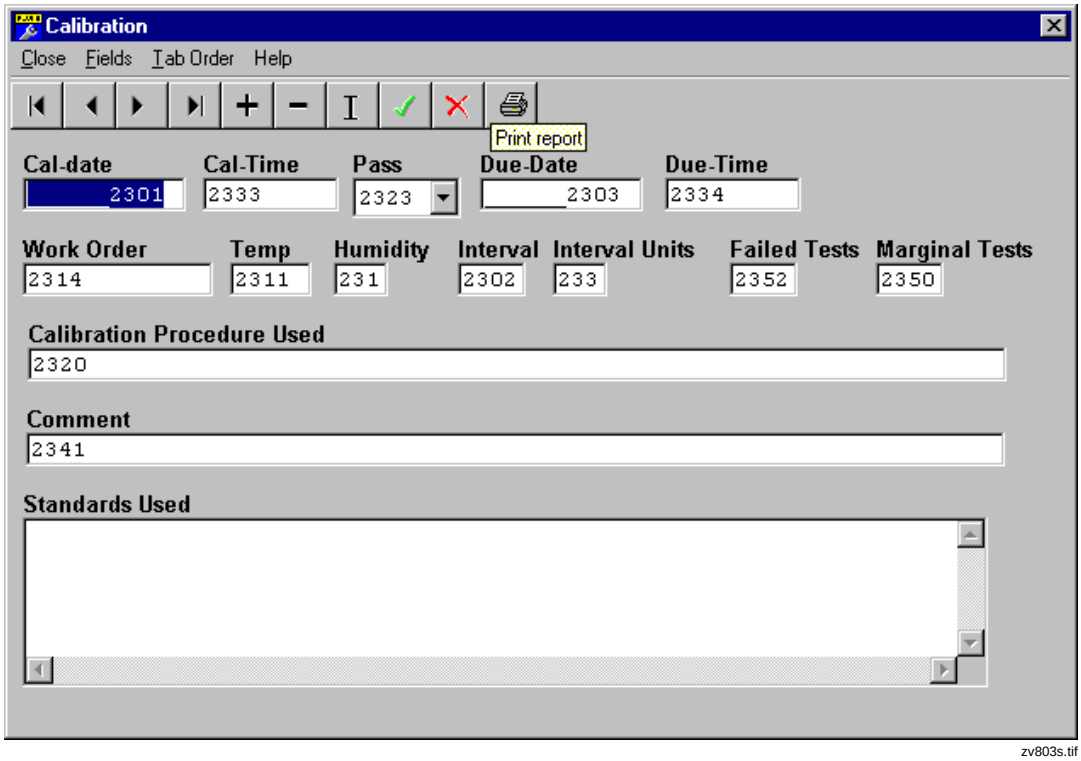


Figure 8-3. Calibration Form

The fields necessary for the calibration form to work properly are those labeled Cal-Date, Cal-Time, Pass, Interval. These fields represent the minimum amount of information that the database needs, and ensures that you will always record the date and time the calibration was performed, whether the asset passed or failed, and the calibration interval.

Core fields are required only for the proper functioning of the Fluke Metrology application programs. They are not necessary in the sense of returning useful information. It is true that the database will not return any useful information if you do not supply it in the first place, but it is also true that to customize any of the forms you must be aware of the significance of the core fields the form contains. You might be particularly interested in some piece of information that is not a core field, or perhaps is not even represented on the form. And in fact, you can use any of the non-core fields to keep track of completely unrelated data if you wish.

You can do that because the important facts about the record you create are being stored in the core fields. You can customize a form as radically as you want; even to the point of having none of the core fields visible. As long as the fields are receiving data from somewhere, everything will sail along smoothly.

For example, one customizable field attribute is “copy forward.” This attribute lets you specify another field to copy data into. In order to suppress the core field Cal-Date, you would have to ensure that some other field is supplying the data. Otherwise, the system will complain.

The underlying design supports your efforts to keep records of all the important data about every calibration event, whether you record the information automatically as a consequence of running a Calibration Procedure, or whether you are entering data manually. A major consideration about customization is that you must be aware of the underlying design in order to change it.

The following list shows the minimal fields on the data entry forms:

Calibration Form		**Inventory Form**	
2301	Calibration Date	4201	Asset Number
2333	Calibration Time	4209	Trace
2323	Pass/Fail	4299	State
2302	Interval (from 4229)	4240	Segregate value
2331	Interval Type (from 4228)	4228	Interval Type
2316	Arrive (from 4224)	4229	Interval

In every case, the title of the field before customization reflects the recommended use for that field within the philosophy and workflow process outlined in Chapter 1.

Calculated Fields

In the database, a calculated field is one that depends on the value in another field or one that supplies a value to another field. This value can be a direct copy, as in the case of 2302, 2316 and 2331 on the Calibration form (Interval, Interval Type

and Arrival Date), all of which receive their values from the Inventory form; or the inter-relationships among the fields can be more complicated, as in the case of Calibration Labor Rate calculation.

Using the Calibration Labor Rate as an example, the value in field 2309 (Calibration Labor Cost) is calculated from the values in several other fields, namely 2307 (Calibration Tech), 2313 (Calibration Shop), 2310 (Shop rate)2324 (Calibration Time), 2327 (Second Calibration Technician), and 2328 (Second Calibration Technician Time). The relationships between these fields are best illustrated by the formulas used in deriving the values:

Tech1cost = 2324/caltime1 X Labor rate for 2307/caltech
Tech2cost = 2328/caltime2 X Labor rate for 2327/caltech2
ShopCost = 2310/caltime X Rate for 2313/calshop
2309 (CalLaborCost) = Tech1cost + Tech2cost + ShopCost

It is imperative that any customization of calculated fields take into account the underlying connections between fields that interact. For example, if you were to decide to use field number 2324 to record some information other than a period of time used by the primary calibration technician to calibrate the asset, you would also have to disregard the result of multiplying this time by a labor rate. In other words, the fact that these fields interact and that their values are interdependent means that any customization must either disregard the calculation, or make use of it in pretty much the same way as the original relationships demand.

Another significant matter concerning calculated fields is the effect customizing them has on the sample reports supplied with the software. Here again, if you do not stay mindful of the underlying relationships between the fields, you could easily generate confusing or meaningless reports.

The following are calculations that the Fluke Metrology software performs and the fields involved:

Due Date Calculation

2301	Cal Date	2323	Pass/Fail
2302	Cal Interval	4228	Interval Type
2303	Cal Due Date	4229	Master Interval

Calibration Labor Rate

2307	CalTech1	2327	CalTech2
2313	CalShop	2328	CalTime2
2324	CalTime1	2310	CalTime

Total Calibration Time

2310	Total Cal Time	2324	Cal Time
2307	Cal Tech 1	2327	Cal Tech 2
2313	Cal Shop	2328	Cal Time 2

Repair Labor Rate

3233	Rep Tech 1	3237	Rep Tech 2
3236	Rep Shop	3738	Rep Time 2
3239	Rep Time 1	3229	Rep Time
3235	Labor Cost		

Total Repair Time

3229	Total Repair Time	3239	Repair Time
3233	Repair Tech	3237	Repair Tech 2
3236	Repair Shop	3238	Repair Time 2

Repair Due Date

3230	Repair Date	3249	Repair Interval
3247	Repair Due Date	3250	Repair Interval Type

Total Parts Cost

3209	Cost of Part 1	3218	Cost of Part 4
3212	Cost of Part 2	3221	Cost of Part 5
3215	Cost of Part 3	3224	Cost of Part 6

Note

Some fields are considered to be "calculated" if they are subject to rules enforced by the database. For example, As Found/As Left can influence the values recorded in the Reliability field, 2304.

For specific information about how the fields interact and the formulas used in the calculations, refer to the on-line help.

Auxiliary Fields

Fluke Metrology software has the potential to be molded to fit just about any metrology lab's needs. Therefore, many of the fields are neither core nor calculated, they're just there.

You can pretty much use an auxiliary field in any way you want, consistent with its maximum character length and data type. Remember though that, if you change a field title on a form, you should make the corresponding change on any report you will be using that uses data in that table.

There are many examples of auxiliary fields - most of the Inventory Form is made up of them. It is not necessary, for example to use 4202 for the name of an asset's manufacturer. If you need a 20 character field for something else and 12 characters are sufficient for your list of manufacturers, you could use 4210 for that, and in this way free up the 20 characters in 4202.

When you change the usage and title of a field simultaneously, you may want to document your customization. It is not strictly necessary, though, since you can

easily generate a report that shows your current customization, or use the field editor to check out the current situation. The field editor is the next topic.

Customizing Fields and Forms

Forms provide a focal point of contact between the database and a user. Through customization, you can make the forms more familiar and useful by using terms in everyday use at your installation, rather than having to change your terminology to conform to the default usage.

For example, here at Fluke we always call each instrument within a metrology system an ASSET, and so we use the ASSET NUMBER as the primary number for tracking and managing the instrument. At your site, you may have standardized on calling your gear a PROPERTY, INSTRUMENT or some other term. If your user encounters a form asking for an ASSET NUMBER, there could be some confusion. By being able to change the form, you ensure good communications through consistency and clarity. If you were unable to change the term, you might end up using both – a situation that could lead to confusion.

Similarly, your own printed forms or other systems already in place may have a standardized order or appearance. If you always ask for the instrument type before the model number, for example, it would seem odd if you had to change this expectation just because that is the way the Fluke software does it.

The fields can be placed anywhere on the form, or you can take any field off of a form. Whatever fields remain can then be re-ordered so that the tabbing sequence between them is logical and predictable.

Customizing Field Properties

Almost every property of every field can undergo customization. There are only a few exceptions, because the Fluke Metrology software has been designed to provide maximum flexibility, and customizing fields is an important ingredient in this design.

Field types (the kind of data that the field can contain) and the numerical relationships that bind calculated fields to one another are fixed, since these are fundamental to the operation of the software. However, field titles, lengths, and their use are completely open to your discretion, as are their locations on forms. Repositioning fields on forms is discussed a little further on in this chapter.

To change a field's properties, proceed as follows:

1. Start the DB Setup Application, logging into the data source as MT.

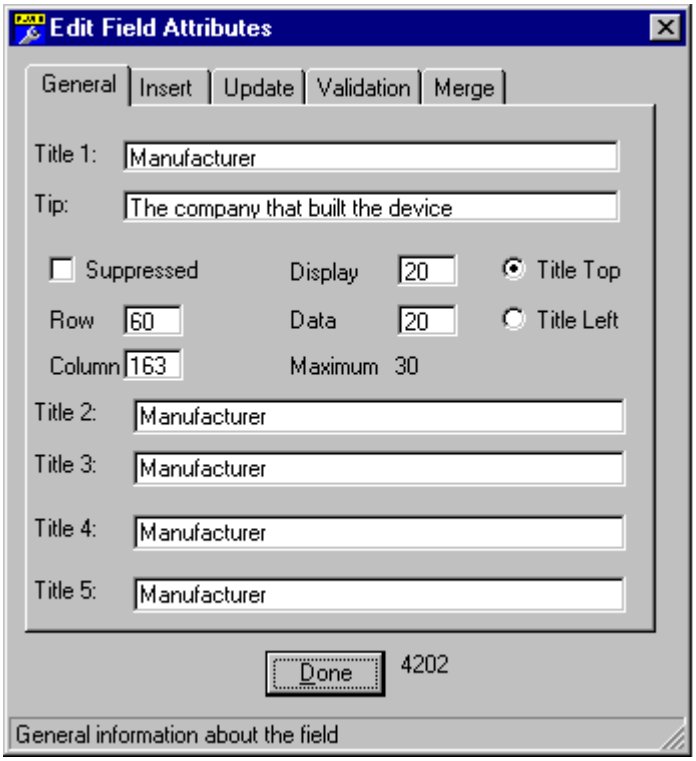
   Only the MT user is permitted to use the DB Setup Application.

2. When the DB Setup Application starts up, select **Fields** from the **Customize** Menu.

   This displays a listing of all the fields.

3. Select the field you are interested in.

   A special field editor displays, showing the current settings for the field you have selected. Refer to Figure 8-4.



zv804s.bmp

Figure 8-4. Edit Field Dialog Box

This example shows the field editor as it would appear for Field 4202 before any customization has been done. The following paragraphs explain the selections possible in the field editor.

General Tab

The selections within the tab labeled “General” allow you to change the field’s title, and to select whether it is displayed to your users on forms. “Suppress” means that the field will not be displayed to users. Even though it is not displayed, a field can still be used.

Positioning

This box changes the physical position of the field on the form. The physical location is specified in grid units relative to the upper left corner of the form.

Length

The field can be adjusted within the maximum field length indicated. The maximum length is not adjustable. However, you can change how much of the field length you actually use by changing the value indicated as “Data”, and you can change the length that will be visible on the form by changing the number indicated as “Display.” If the Display length is shorter than the Data length, the field’s window will scroll.

Update Tab

These selections alter the access levels that are permitted to change records, establish whether or not the field is required, set the default value for the field (if any), and determine whether field copying is activated for this field.

REQUIRED FIELDS
Some fields are always required. Asset Number (4201)is one. Others can be customized to be required, too. *Required* means that the field must have data in it when the record is saved. If the user does not have the proper access level for a required field and the field is blank, the record cannot be saved.

You can set up a default value or use **Source Copy** to cause the field to get data. Either of these satisfies the requirement.

ACCESS LEVELS
If a field is required, the record cannot be saved if the user does not have the proper access level, since the user has no way to enter data into the field. Set the allow level to the Access Levels you wish to permit to change the data.

Insert Tab

These fields allow a different access level and default value than those that follow. This permits you to restrict users who would otherwise have permission to change the field from changing the first record. The selection box “Required” makes it a requirement that there be a value for the field.

Source Copy / Destination Copy

Inventory fields can be pre-loaded with information from another field. Inventory data contains no history, but you can use this selection to copy the value from one of the history forms into the inventory form.

For example, assume you want to have the current Due Date appear in the Note-date field on the inventory window. Use the customization editor to enter 2303 (Cal Due-date field number) as the “Copy from field” number for the Note-date

field. Now every time a new calibration record is saved the due date will be copied from the calibration record to the Note-date field.

When you are creating a new Cal, Loc, or Maintenance record (history records), and if you had enabled Copy Forward for one or more fields, the previous values in those fields are inserted into the record you are creating.

If you have enabled Copy From for any of the History records (Cal, Loc, or Maintenance records), then, when you save the History record, the Inventory record is updated immediately.

Any field can have its value copied forward to the next record (the record you are creating). Note that if you are creating a new inventory record, the value is not available to a query until after you have saved the record

Field by field, you can choose which values will be copied forward to the next record. Copy forward takes precedence over Default data. If a field has both default data and copy data, the copy data appears in the field, not the default data.

For fields that contain historical data (Calibration, Location), legal values are ON and OFF. The customization editor restricts your choice to one of these two values.

If you set copy forwarding to ON, then the entry from the previous record is brought forward when a new record is added to the system. For example, you may wish to copy the Temperature field to a new calibration record. By setting the copy value to YES, the temperature from the previous calibration record will be copied to a new calibration record. This value can be changed if the user has sufficient authorization for the temperature field.

If you turn copy forwarding OFF, then there is no effect on the new record.

By setting the initial default value to blank and making the field required, you can ensure that the field is filled the first time it appears in a record.

Default Field Attribute

Default data is that which appears automatically in a field on a data entry window or during importation. Default data can be specified for both Insert and Update data entry.

Default data overwrites any data that is currently in the field with the exception of data caused by the Copy attribute being active. Default data can be changed if the user has authorization for the field.

There are three special default data constructs, as follows:

“	cause an alphanumeric field to be blanked.
99/99/99	causes today’s date to appear in a date field.
99:99	causes the current time to appear in a time field.
U_SEG	prefills the field with the user segregate

U_WRK	prefills the field with the workstation ID
U_NAM	prefills the field with the user's ID

Validation Tab

Validation provides a way to restrict the data allowed in a field to a set of authorized, or valid, values. During normal operation, the cursor does not move from a validated field with a bad entry until it is corrected.

You create and edit valid entries for a field within the DB Setup Application's field editor. Access it by clicking the **Validation** tab.

Note

In any large database, valid data entry is very important. The entries you type in the DB Setup Application are used for validating the field data. Careless editing can result in the applications accepting incorrect data or rejecting correct data.

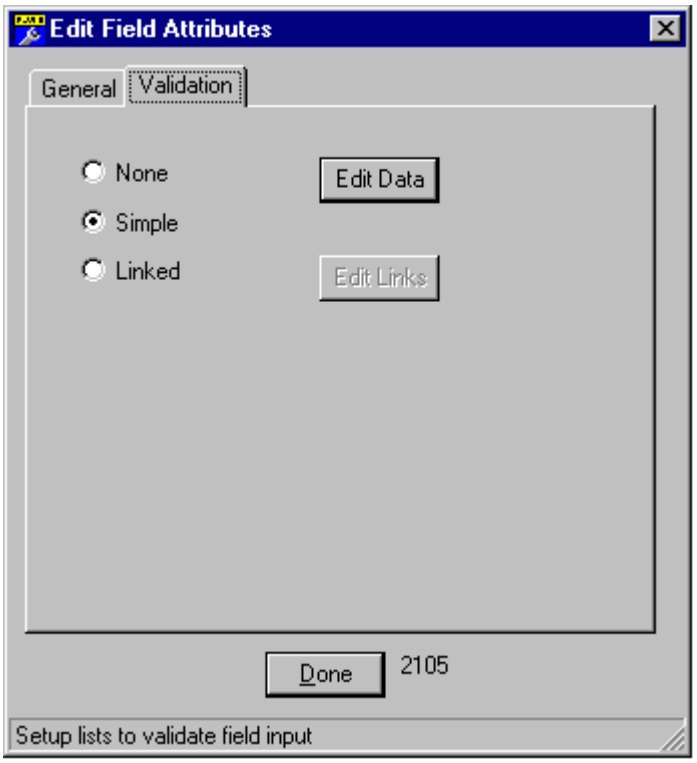
Try to keep the number of fields you validate manageable in size, and do not set up a field with a large number of valid entries. For example, if you are validating half to two-thirds of all of a form's fields, that is probably excessive; if you have more than 50 valid entries for a given field, that is quite a lot, too.

There are two formats for validation entries: simple validation and linked validation. These are explained in the following paragraphs.

Simple Validation

To edit field validation, complete the following steps.

1. Open DB Setup.
2. Select **Customize**, then **Fields**.
3. Select the field you wish to edit from the list provided.
4. In the Edit Field Attributes dialog, press the Validation tab. The following screen appears.



za805s.bmp

Figure 8-5. Field Validation

5. Select Simple Validation to enter the Edit Validation dialog box.
   - You can enter data manually by using the plus, minus, and check buttons on the tool bar.
   - You can also get the data that is currently in the database by selecting "current data." If you already have data, this is a good way to get started. You can then add or edit entries.
   - You can also get data from a file. The file must be an ASCII text file, one entry per line.

Linked Validation Entry

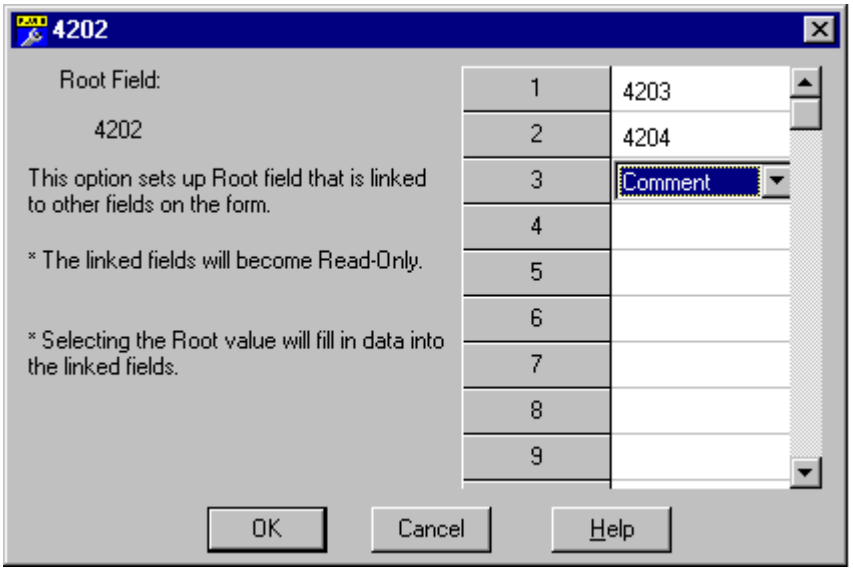
Linked validation is a powerful feature allowing information to be entered into multiple fields at one time. One base field provides the information for other, linked fields. The validation data is associated with the base field, and is filled into the linked fields when the operator exits the base field.

Create linked validation by a two-step process. First, select the fields to be used. Then, enter the data.

To create linked field validation, complete the following steps.

1. Open DB Setup.
2. Select **Customize**, then **Fields**.
3. Select the field you wish to edit from the list provided. We'll use field 4202 as an example.
4. In the Edit Field Attributes dialog, select the Validation tab.
5. Select Linked, then Edit Links.

The following screen appears:



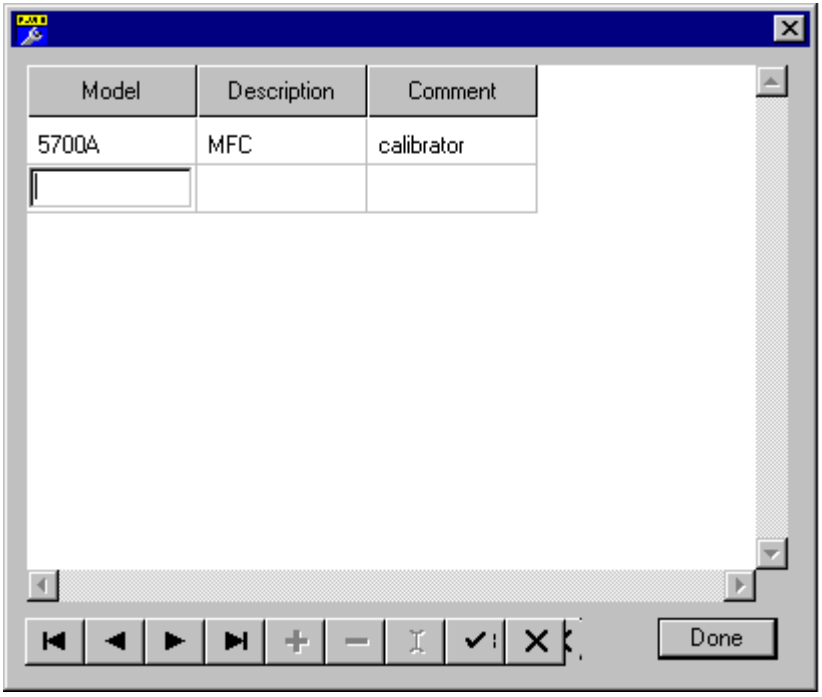
zv811s.bmp

Figure 8-6. Edit Links Dialog

In this example, field 4202 is the root field. Double-click on the highlighted box to bring up a drop-down menu of possible links. We have chosen to link field 4203 and 4204 to field 4202. Field 4203 and 4204 will not be editable. We have also chosen to have a comment.

Once you have chosen fields for linked validation, select **OK**, then **Edit Data**.

The following screen appears:



zv812s.bmp

Figure 8-7. Edit Data for Linked Validation

Press the + on the tool bar to add new data. Enter the data you want to fill into the linked fields when you select the value for the root field (in this case, 4202) in the manual entry tool.

Merge Tab

In DB Setup, access the merge feature by selecting **Customize**, then **Fields**. Select a field to edit. The Edit Field Attributes dialog appears. One of the features that can appear on the tabs in this dialog is Merge.

The merge feature lets you create your own fields, enter and modify data into those fields, and use their data in reports.

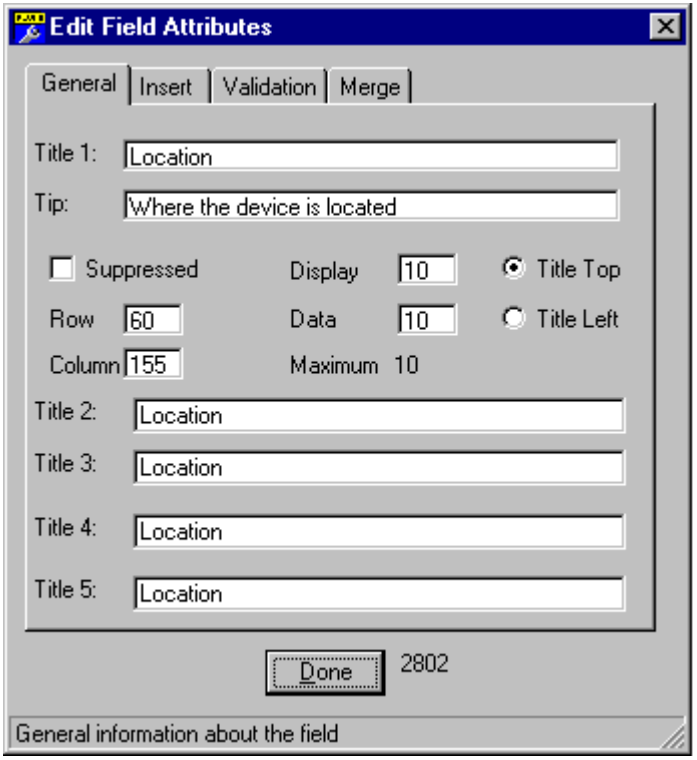
Rules of the Merge Feature

- Only users with access levels 3, 4, and 5 can add or modify merge data.
- Users with access levels 2, 1, and 0 can only view the merge data.
- Merge data for any given field can be a maximum of 80 characters, and must be alphanumeric (string) data.
- Any given field is limited to 15 items of merge data.

- Only certain fields can have merge data associated with them. If the field cannot have such data, the Merge button is grayed out in the Edit Field Attributes dialog.

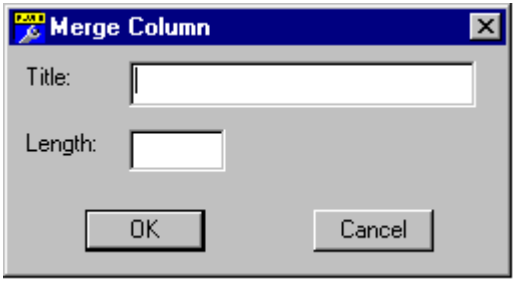
Creating New Merge Fields

1. Start the Calibration Database engine, if it is not already running.
2. Start the DB Setup application from the Fluke Metrology Software program files.
3. From the menu, select **Customize**, then **Fields**.
4. From the list, locate the field to which you wish to add merge data and click the **Edit** button.
5. When the Edit Field Attributes dialog appears, make sure the Merge button is not grayed out. If it is, the field is not available for merge. In this case, select another field for which the button is available. The following field shows the Edit Field Attributes dialog for Field 2802, Location.



zv806s.bmp

6. Click the Merge tab. The next window has two buttons labeled **Off** and **On**. Click the **On** button and then click **Add**. The Merge Column dialog box appears.



zv807s.bmp

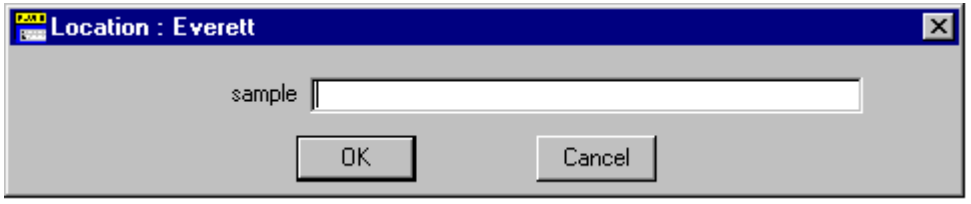
7. You can now enter the new field title and length or modify an existing entry.

Entering Merge Data

1. Enter data for merging as described in the following steps.
2. Start the Calibration Data Engine, if it is not already running.
3. Start the Manual Entry application from the Fluke Metrology Software program.
   a. To modify merge information for an asset that already exists, but that you must search for, select **Search**, then **Find Asset**.
   b. To modify merge information for an asset when you already know the asset number, select **Search**, then **Old Asset**.
   c. To create or modify information for an asset that does not yet exist in the database, select **Search**, then **New Asset**.
4. Once you see the inventory form for the asset you have chosen, go to the menu and select **Form**, then **Location**.
5. Click the **+** button to activate the window, as shown below.

zv808s.bmp

6. Place the cursor on a field that you have previously set up with a merge table. In this example, that would be the Location field. When the cursor is in that field, go to the menu and select **Field**, then **Merge**. The Merge dialog appears.



zv809s.bmp

7. If the merge table does not yet contain any data, all the fields are blank. Otherwise the values are shown in the Merge window.

Using Merge Data in a Report

When you set up merge data, you actually expand the database. You add a table and a number of fields (up to the maximum of 15 for any given root field). The table you create can be used in reports, just like any other table. Besides the rules given at the beginning of this discussion, there are a few points to note about the merge tables you add:

- When you create merge data, you are actually creating a new table in the database, called the merge table. The name of this table is always *mt.mrg<field number>*, where <field number> is the root field number.
- The first field in the merge table is always named “merge.” This field cannot be blank, must be unique, and must be the same size as the root field.
- When used in a report, the merge table must be specifically linked to the root field. This is done using Crystal Report’s Visual Linking Expert.

Customizing Help

To create a custom help file for the fields that you have modified, proceed as follows:

Use Notepad or a similar text editor to create a file called MT.HLP. In this file, each field you wish to document has a separate entry.

Field entries in the MT.HLP file begin with an asterisk character (*), and then the field number. This introducer must be on a line by itself.

Following the field introducer, include any text you wish to appear for your users.

```
*2301

This text contains your customized information about
field 2301, Calibration Date.

*2302

This text contains your customized information about
field 2302, Calibration Interval.
```

Save MT.HLP in the same directory as your MET/TRACK executables (by default, \metcal\ on the shared drive.)

To access the customized help, highlight a field using the TAB key, then press Ctrl F1.

Repositioning a Field on a Form

There are two ways to change the location of a field on a form. Within the Edit Field dialog box just described, you can change the Row and Column numbers, or you can move the fields around using the cursor. A combination of the two techniques gives the best results. To use the visual method of moving the fields, proceed as follows:

1. Within the DB Setup Application, select **Forms** from the **Customize** Menu. This displays a submenu asking which form you wish to work on.
2. Choose a form. The software displays a replica of the form in a special forms editor. To move a field's location, hold down the **right** mouse button anywhere within the field.
3. Drag to the desired location for the field and click again. The field moves to the new location.
4. Move those fields that you want to have in different places, then close the form. Changes are saved automatically.

Note

For best results, use Customize Forms only for roughly locating the fields, then do fine-tuning using field customization. Once you have the fields roughly lined up, adjust the Row and Column values in the Edit Field dialog to position them exactly. Refer to the discussion of the Edit Field dialog box for complete details.

Customizing the Tab Order on a Form

After you have changed the appearance of a form, you may also want to modify the tab order so users can move from one field to the next in a logical sequence. If you have moved some fields around, and suppressed others so they are not even visible to the user, the final form will be greatly enhanced if tabbing between fields moves the insertion point logically.

To change the tabbing order on a form, proceed as follows:

1. Within the DB Setup Application, open the **Customize** menu, select **Forms** and open the form on which you want to change tabbing order. From that form menu, select **Tab Order.**
2. A Tab Order dialog box for that form appears. You can now click and drag the field numbers into any order you like. Only the field that can be edited appear in the list.

Other Administrative Tasks

As stated in the introduction to this chapter, not all the administrative tasks are concentrated in the DB Setup and Run Time applications. The following list outlines some of these tasks, and shows where to look for information.

Adding or purging assets in the database	Manual Entry Application, Chapter 2
Adding a report that has been created or modified using Crystal Reports	Manual Entry Application, Chapter 2
Creating a new report, or modifying an existing one	Reports, Chapter 5
Creating, modifying, or testing a calibration procedure	Editor application, Chapter 4
Changing the system's behavior	Control Files, Chapter 6
Deleting a calibration record	Manual Entry Application, Chapter 2

Appendices

Appendix A
How to Access MET/SUPPORT™

About the MET/SUPPORT Program

Fluke Metrology Software is supported by a highly skilled and conscientious team of professionals who are available to answer questions, help solve unique problems, and assist in troubleshooting in the event you run into difficulties.

Telephone Support

Free telephone support is available, but only to registered users of Fluke software. Be sure to register your software as soon as possible so you can make use of this service should the need arise.

Within the United States, toll free telephone support is available from 8:00 AM to 4:30 PM, Pacific Standard Time.

Phone: 1-800-825-7411

Mail, FAX, and World Wide Web

You may also contact us by mail, FAX, or visit the FLUKE World Wide Web site.

FLUKE Corporation	**Phone**	1-425-356-5870
PO Box 9090	**FAX**	1-425-356-5992
Everett, WA 98206-9090	**E-Mail**	metsupport@tc.fluke.com
Attention: MET/SUPPORT, M/S 275G	**WWW**	*http://www.fluke.com*

Before Calling Technical Support

You may be able to discover the answer for yourself., so take the following steps before calling for support:

- Check the manual
- Check the on-line help
- Check the FLUKE World Wide Web site

Please Have the Following Information Available

If you do find it necessary to call for assistance, take a few moments to jot down some information. If this information is readily available, the support person can more quickly get to the exact details of the difficulty you are having, and will be able to resolve the issue more efficiently.

- Product name, applicable licenses, version and serial numbers
- Manual title, part number and version
- Operating system and version
- Network information if applicable
- List of steps necessary to recreate the problem
- Contents of autoexec.bat and config.sys files if applicable
- Version of *Crystal Reports*

International Support

For more information on international MET/SUPPORT™, please contact the Fluke office nearest you or your local representative.

	TELEPHONE	FAX
Fluke Canada	905-890-7600	905-890-6866
Fluke China	86-10-6 512-3435	86-10 6 516-3437
Fluke Europe Headquarters (NL)	31-40-2678200	31-40-2678222
Fluke France	33-1-48-96 6300	33-1-48-96 6350
Fluke Germany	49-561-95940	49-561-9594-119
Fluke Japan	81-3-3434-0181	81-3-3434-0170
Fluke Singapore	65-276-5161	65-276-5759
Fluke UK	44-923-240511	44-923-225067

Appendix B
Computing Test Uncertainty Ratio

Computing Test Uncertainty Ratio

The Test Uncertainty Ratio (TUR) is computed only after compilation or execution of a result-generating instrument FSC and the MEMC FSC. This computation is done both during the final pass of compilation and while executing the procedure. At compilation, the Test Tolerance is determined as the distance to the closest tolerance limit (unless one of the limits is zero.) For example, a +3U in the TOLERANCE field (implying -0U tolerance) will cause the Test Tolerance to be 3U. At execution, the TUR is calculated as the distance from the specified Nominal to the tolerance limit which falls on the same side of the Nominal as the System Actual.

Note

MET/CAL gives a warning when the TUR drops below tur_lim. The compilation still succeeds. For example, a test can specify that a 1% meter should be perfect. The test tolerance is therefore zero. Even a 10 ppm calibrator is not good enough to avoid the creation of a warning, since the TUR is also zero.

All accuracies of calibration instruments are symmetrical around the programmed nominal value. However, the tolerance of the test can specify different tolerances in the positive and negative directions. In that case, for the purposes of computing the TUR, the tightest tolerance is used. This means specifying the tolerance as +0.1% implies the negative tolerance is 0. This always results in a TUR warning of 0 that normally can be ignored.

Example TUR Calculation

```
UUT Indicated = 100V
Tolerance     = -2U +1U
System Actual = 99V
```

The test tolerance on the lower side of the UUT Indicated is used because the System Actual is less than the Nominal. The test tolerance on the lower side of 100V is 2V, so the ERROR%TOL is 50%.

Instrument Evaluation Test with Known Nominal Value

The calibration system uncertainty is obtained from the information in the calibration instrument related Accuracy File.

The allowed tolerance for the test is derived from the procedure line. In the most common situation, the evaluation step appears in a fully defined test; for example:

STEP	FSC	RANGE	NOMINAL	TOLERANCE	MOD1	MOD2	3	4	CON
2.001	5700	10	10V	0.1%	0.1%				2W

The allowed tolerance is 0.1% of nominal (10V) and 0.1% of range (10), so the total tolerance is 0.02V. The TUR is:

test tolerance / calibration system uncertainty

If the TUR drops below tur_lim, the calibration system reports the following as a warning during compilation:

- The absolute value of the test tolerance.
- The absolute value of the calibration system uncertainty.
- The Test Uncertainty Ratio.

If the TUR drops below tur_lim, the calibration system adds to the test result the value of the TUR.

Note

All TURs are computed in linear scale units. Any nominal values or Accuracy File information expressed in dBm are first converted to volts. A warning includes the linear values.

The Test Uncertainty Ratio of a Comparison Test is computed using the stated Nominal value. It is possible for erroneous warnings to result from this assumption if the value stored in MEM and the NOMINAL field differ substantially.

Instrument Evaluation Test with Unknown Nominal Value

If the nominal value is not defined in the procedure, it is difficult or impossible to determine the TUR when the procedure is compiled. However, during calibration procedure execution, the TUR can be defined since the nominal value is known. Therefore, the reporting of the TUR being too low is the same as previously described.

During procedure compilation, the system attempts to estimate a worst case TUR for the test. The worst case occurs with the worst tolerance of the calibration system instrument and the minimum tolerance of the test. It is not possible to verify that the conditions assumed are indeed worst case, although they typically are. Some estimate can be made if at least the range is specified. Without that information, no TUR estimate is possible.

The allowed tolerance for the test cannot be totally derived from the procedure line. For example:

```
STEP    FSC   RANGE NOMINAL    TOLERANCE     MOD1    MOD2  3  4 CON
2.001   5700   10    V          0.1% 0.1/                      2W
```

The allowed tolerance is 0.1% of nominal (the nominal value to be taken from memory register MEM) plus 0.1% of range (10), which amounts to 0.01V. In this case, the TUR is computed as follows:

$$\text{The TUR} = \frac{\text{test tolerance with Nominal set to the Range value}}{\text{calibration system uncertainty with Nominal set to Range value}}$$

If this TUR value drops below tur_lim, the calibration system reports the following as a warning:

- Estimated absolute value of the test tolerance at the Range value.
- Estimated absolute value of the calibration system accuracy at the Range value.
- Estimated Test Uncertainty Ratio.

MEMC FSC with Known Nominal Value

The MEMC FSC is used most often in closed loop calibration using a Setup FSC, followed by an IEEE FSC, a MEME, and a MEMC to determine the outcome of the test. For example:

```
STEP   FSC   RANGE NOMINAL  TOLERANCE   MOD1   MOD2  3  4 CON
2.001  IEEE        F1R3
2.002  5700        10.00V                              S  2W
2.003  IEEE        ?[I]
2.004  MEME
2.005  MEMC   10    V       0.1% 0.1/
3.001  ....
```

The calibration system uncertainty is therefore generally determined by the last instrument Setup FSC preceding the MEMC FSC. In this example it is step 2.001: the Setup test of the 5700 FSC. Whenever a Setup FSC is encountered, the MET/CAL compiler stores the calibration system uncertainty of this Setup test using the information of the Accuracy File of the related instrument, in this case the 5700A. This accuracy is used for computing the TUR when the MEMC FSC line is being verified during compilation or executed during procedure execution.

The allowed tolerance for the test is derived from the MEMC procedure line. According to the example, the allowed tolerance of the test is 0.01% of the nominal value (MEM1) plus 0.001% of the range (10) so that the total tolerance is 0.011. In this case

The TUR = test tolerance/stored calibration system uncertainty

If the TUR drops below tur_lim during procedure compilation, the calibration system reports the following as a warning:

- The absolute value of the test tolerance.
- The absolute value of the calibration system accuracy.
- The Test Uncertainty Ratio.

During test execution, if the TUR drops below tur_lim, the calibration system reports the value of the TUR.

Note

Erroneous warning messages can be generated when the last Setup test is not the one that represents the calibration system accuracy.

MEMC FSC with Unknown Nominal Value

A MEMC FSC with an unknown nominal value is similar to the case of an instrument FSC with an unknown value. When you execute the procedure, the nominal value is known, and the TUR is reported, if it is below the tur_lim value. During procedure compilation, a TUR may be estimated or an estimate may not be possible.

If no range information is provided, an estimate of the TUR cannot be defined during compilation and no estimate is made. Some estimate could be made if at least range information were present. The MEMC line in the previous example now appears as follows:

```
STEP   FSC  RANGE NOMINAL    TOLERANCE     MOD1     MOD2  3  4 CON
2.004  MEMC 10    V          0.1% 0.1/     01/
```

The calibration system accuracy is determined again by the last Setup test in the procedure, as was described earlier where the Nominal value of the MEMC FSC was defined. It is assumed that the nominal value is set to the range value.

The allowed tolerance for the test is 0.1% of nominal (the nominal value to be taken from memory register MEM) plus 0.1% of range (10), which amounts to 0.01V. In this case:

$$\text{The TUR} = \frac{\text{test tolerance with Nominal set to the Range value}}{\text{system uncertainty with Nominal set at the Range value}}$$

If this TUR value drops below tur_lim, the calibration system reports the following as a warning:

- Estimated absolute value of the test tolerance at the range value.
- Estimated absolute value of the stored calibration system uncertainty at the Range value.
- Estimated Test Uncertainty Ratio.

Note

Erroneous warning messages can be generated when the last Setup test is not the one that represents the calibration system accuracy.

Warnings are given only when the TUR drops below tur_lim. The compilation is still successful. For example, a test can specify that a 1% meter should be right on. The test tolerance is therefore zero. Even a 10-ppm calibrator is not good enough to avoid the creation of a warning, since the TUR is also zero.

Note

The variable tol_ref in the metcal.ini file determines if the relative test tolerance computations are based on the UUT_Indicated value (as currently implemented) or the Nominal value (as requested by some MET/CAL users).Most specifications of DMMs are based on their reading, i.e., their UUT Indicated value. In many situations (closed-loop calibrations), this leads to the unexpected result that the tolerance band is not symmetrical around the fixed Nominal value. The tol_ref variable allows symmetrical tolerances around the Nominal value. The tol_ref variable can assume one of two values: tol_ref = UUT_INDICATED or tol_ref = NOMINAL, but only affects the portion of TOL that is ***%*** *or* ***PPM****.*

Appendix C
Table Definitions for Reports

Introduction

This appendix contains information about the database tables and fields for help in creating your own reports, or to add fields to the supplied reports. The illustration in Figure E-1 shows all the tables that you will be concerned with as you create reports.

Run TBLDEFS.RPT to get field information on tables.

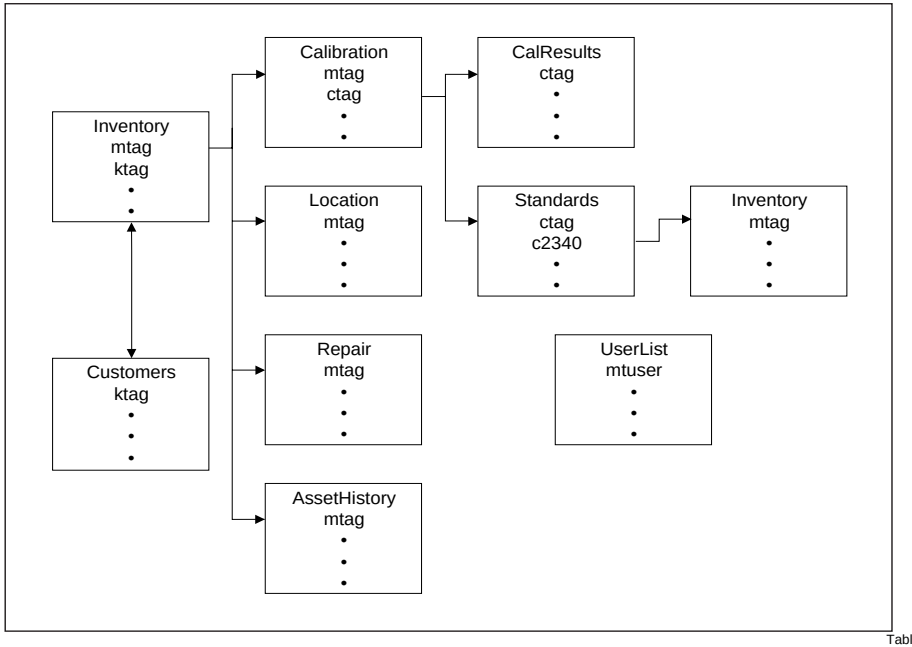


Table Structures

Figure C-1. Tables Used in Writing Reports

Contents of the Tables

The following list describes table contents.

- The ***Inventory*** table contains information that describes every asset being tracked.
- The ***Calibration*** table contains summary information about each calibration event performed.
- The ***CalResults*** table contains individual test results for each calibration if available.
- The ***Standards*** table can contain *mtag* values in field C2340 for each of the standards used during a calibration event.
- The ***Location*** table contains information about where the asset can be found when it is in-use.
- The ***Repair*** table contains information about repairs made to the asset.
- The ***AssetHistory*** table tracks all changes made to asset number in the ***Inventory*** table.
- The ***UserList*** table contains information about every user ever allowed into the system.
- The ***Customers*** table contains information about every customer ever entered into the system.

MET/CAL makes automatic entries in **Inventory**, **Calibration**, **CalResults** and **Standards**.

Rules for Linking Tables when Designing Reports

- Every asset record in the ***Inventory*** table has a unique *mtag* value.
- Link ***Inventory*** to ***Calibration*** history records using the *mtag* field.
- Link ***Inventory*** to ***Location*** history records using the *mtag* field.
- Link ***Inventory*** to ***Repair*** history records using the *mtag* field.
- Every cal event record in the ***Calibration*** table has a unique *ctag* value.
- Link ***Calibration*** to ***CalResults*** records using the *ctag* field.
- Link ***Calibration*** to ***Standards*** records using the *ctag* field.
- Link ***Standards*** to another instance of ***Inventory*** using C2340 and *mtag*
- Link ***Customers*** to ***Inventory*** using ktag.

Various tables can be linked to ***UserList*** to find the real name of the person who made entries and changes to the database. For example, link ***Inventory.I4239*** to ***UserList.mtuser*** to find out who created an inventory record.

Appendix D
Glossary

-A-

Accuracy File

An Accuracy File contains information about the accuracy specification of an instrument associated with an FSC. Unless TUR checking is disabled, there must be one Accuracy File associated with each instrument FSC that is being used in a calibration procedure.

Full details can be found in Chapter 7 of the Fluke Metrology Software Users Manual.

As Found/As Left

As Found/As Left represents a calibration philosophy about how calibration is performed. MET/BASE supports three basic philosophies. Which philosophy you choose determines how the Found/Left field (2308) is used. See As Found/As Left under Special Features in Chapter 8.

asset

A single piece of equipment; each asset is numbered, and the number is used for identifying the asset throughout the database.

-B-

backend data

Any data residing on a server.

backup

It is important to make regular backups of your database files in case of media failure. A backup is a copy of the database file (mtrack.db) and may also include a copy of the

transaction log file (mtrack.log). You can make backups using the SQL Anywhere backup utility or other archiving software of your choice.

-C-

client

The term client is widely-used to mean several different things. Without modification, it may mean for example, "client application", or "client workstation".

Just as with the term "server", the word requires modification, otherwise if may be ambiguous whether it refers to a program or to the computer on which the program is running.

client application

An application that executes on the workstation and interacts with the user.

client computer, client workstation

A computer that accesses shared network resources provided by another computer (called a server). Also referred to as a workstation or client machine.

client/server

A term generally applied to a software architecture in which processing functions are segmented into independent collections of services and requesters on a single machine or segmented among several machines.

One or more processing servers provide a set of services to other clients on the same or across multiple platforms. A server completely encapsulates its processing and presents a well-defined interface for client applications.

For example, MET/BASE is a client/server program. It always runs over some type of network. In a standalone installation, it makes use of a loopback network; the client and server are running on the same computer.

client/server computing

A mode of distributed network computing in which an

application is executed cooperatively by two computers. The client application (the front end) executes on a workstation, while the data engine (the back end or data server) runs on a second computer. (Note that in a MET/BASE standalone installation, both would run on the same computer.)

Client/server computing allows more effective use of computing resources, higher performance, greater flexibility, simpler upgrades, greater reliability and data integrity.

confirm password

Passwords never display, so when you get a new password, you must type it twice. This is a safety measure, and it helps you to remember the new password.

copy data

A configurable attribute of a field that specifies whether the data in the field is to be copied forward to the next record. This field attribute can be used in many different ways to simplify data entry; for example, when first setting up a system, you might set field 4202 (manufacturer) to copy forward while you key in all the assets made by a particular manufacturer.

Correction File

A file that contains information on special calibration

points of the instrument associated with the instrument FSC. An electronic instrument is far more stable during a limited period of time than its stated accuracy indicates. By providing specific calibration points and reducing the calibration interval, the uncertainty of the calibration instrument under those conditions can be significantly reduced. See Chapter 7 of the Fluke Metrology Software Users Guide for details and examples about creating, naming, and using correction files.

CTAG

A special field in the database that serves to link tables to one another so that all the information about a particular calibration event can be retrieved.

The ctag links a calibration event to test results generated during the event and to the standards used.

The ctag is the unique identifier that the software assigns to each calibration event.

Also see mtag and rtag.

customization

The process of selecting operating parameters and defaults to personalize the system for a particular site. For example, the default name of a field may not be the same words as you generally use, so field titles can be changed (customized) to something more familiar to the personnel at the site.

-D-E-

Database Administrator

The database administrator is a person responsible for maintaining the database. The database administrator is generally responsible for all changes to customization, for managing users and their access levels, and for backing up the database.

Also see System Administrator.

database engine

Software such as Watcom or Sybase SQL Anywhere Server that provides high-performance database access to requests for data. Part of the server software package.

data server

Also called a database server; a computer for storing programs and data shared by users on a network. A database server provides database intelligence such as transaction processing, indexing, transaction logging and security, while a file server offers operating system-type file and print capabilities.

default data

A configurable attribute of a field that specifies what text is to appear in the field when the form is opened.

directory

A computer term denoting a subdivision of media that can contain other directories or files. Also called a folder.

-F-G-

field types

Fields are one of the following types:

Decimal	A real number, generally one or two decimal places.
Integer	A whole number.
Alphanumeric	Any sequence of ASCII characters, except for leading or trailing white space characters (e.g., Space, Tab).

Date Always eight characters, two each for day, month, and year, plus two slashes for separating the numbers.

The style is configurable in the Database Setup application. The three selections available are:

MM/DD/YYYY

DD/MM/YYYY

YYYY/MM/DD

Time Always eight characters, two each for hours, minutes, and seconds, plus two colons for separating the numbers. Times can range from 00:00:00 through 23:59:59.

Boolean Boolean type fields accept a Y or an N meaning YES or NO.

file server

A computer for storing programs and data shared by users on a network. A file server offers operating system-type file and print capabilities, while a database server provide database intelligence such as transaction processing, indexing, logging and security.

Also see server, client.

focus

In working with forms and tables, a field is said to have focus if it is currently the field that can be edited, or have information placed into it. Focus is always on one of the fields of a form, and there is usually some visual indication of the field having the focus, such as the location of a cursor, or sometimes the field with focus has a small dashed border in addition to its regular border. Most Windows applications make use of this device.

folder

A computer term denoting a subdivision of media that can contain other folders or files. Also called a directory.

forward traceability

The term “traceability” or “forward Traceability” refers to the ability of a calibration laboratory to trace an instrument's calibration credibility back to a national, recognized, legal standard.

The first listing in a traceability report shows the standards against which the subject asset was calibrated. The next level are the assets that participated in the calibration of those assets, and so on.

A forward traceability report (also called a "trace up" report), shows the traceability tree of an asset to any given depth, up to the software's maximum.

Also see traceability and reverse traceability.

FSC

Function Selection Code, a keyword in the MET/CAL procedure language. The FSC instructs the Run Time system to perform a task requested by a procedure.

For example, the 5700 FSC instructs the system to provide a stimulus output from the Fluke 5700A Calibrator. The DISP FSC directs the system to display a message to the operator during the execution of a calibration procedure.

See the MET/CAL Reference Manual for details.

-H-

history

A database contains information about an asset that changes over time. There is only one Inventory record per asset, but there may be any number of history records. The three types of history records are the Calibration, Location, and Maintenance (Repair).

-I-J-K-

import, importation, importing

In a general sense, the word "import" means to bring something from one environment into the current environment. Used in relation to Fluke Metrology Software, the word refers to the process and the specific steps one must take to move data from some other system into the Fluke MET/BASE database.

Contrast to migration, which is the one time operation of bringing DOS MET/TRACK data into MET/BASE. The importing of data may occur any number of times, and may even be a scheduled activity at the site. ASCII delimited data from other sources can be brought into MET/BASE at any time.

If your data exists in another database, i.e., not gathered using a Fluke database product such as MET/CAL or MET/TRACK, you must import the data into the database. Importing data is not complicated, but it may require close attention to detail, since metrological traceability is at stake.
Also see migration.

initial data entry, initial access

Initial access or initial data entry refers to the first time that the Inventory data is entered (a new asset), or the first time that Calibration, Location, or Repair data is entered (the first calibration).

-L-

length

The horizontal dimension. In MET/CAL and MET/TRACK forms, the number of characters that a value can contain.

License

A software license is the proof of your agreement with Fluke regarding your proper use of our Software.

A software license constitutes a legal agreement between a purchaser and a software provider. This agreement typically covers such topics as the number of copies you can legitimately make, onto how many systems you can load the software, as well as other issues concerning the legal usage of the software.

Fluke Metrology Software is licensed. It is the purchaser's responsibility to understand and comply with the terms under which the software was purchased.

linked validation

Feature allowing multiple fields to be entered simultaneously. Validation data is associated with a base field and is entered into the linked fields automatically. In the customization editor, the validation entry is the name of the base field. Linked fields are indicated by the name of the base field, a colon, and the number of the column from which data is to be taken. See the printed manual for complete information.

Local Area Network (LAN)

Two or more computers connected to a network that is limited to an area such as a single office building and does not extend across public rights-of-way.

-M-

migration

Migrating data is similar to importing it, but with significant differences.

Importing data implies bringing it from a source other than Fluke Metrology Software. On the other hand, migrating data just means bringing in data from a DOS or Legacy version of MET/CAL or MET/TRACK.

If you gathered your current data using DOS MET/CAL or Version 4 in Legacy Mode, or DOS MET/TRACK, you bring it into the new database by migration.

Also see importation.

MTAG

A special field in the database that serves to link tables to one another so that all the information about a particular calibration event can be retrieved.

The mtag links an asset to the Calibration, Location, and History tables. Any time you create a new asset in the Inventory table, the software automatically assigns a new mtag link so that reports can locate other information about that asset.

Also see ctag and rtag.

-N-

network driver

A network driver is the software logically located between the operating system and the network card, which allows applications to communicate to the network card.

network operating system

An operating system optimized for server deployment, such as Microsoft Windows NT Server Edition. Typically used on the file server and/or data server.

normal data entry, normal access

Normal access or normal data entry occurs any time the Inventory data is changed, or any time (after the first time) the operator accesses Calibration, Location, or Repair data.

network operating system

An operating system optimized for server deployment, such as Microsoft Windows NT Server Edition.

NULL

The NULL value is a special value for a database entry, different from any other valid value for any data type. The NULL value represents missing or inapplicable information.

-O-

old password

Changing a password requires you to know the original password. This is a security measure to protect the data that you put into the system. If you have forgotten your password, you cannot get a new one using this menu. You must have a new password assigned by your Administrator.

Open Database Connectivity (ODBC)

The Open Database Connectivity (ODBC) interface, defined by Microsoft Corporation, is a standard interface to database management systems in the Windows and Windows NT environments.

ODBC driver

A dynamic link library (DLL) used to connect a specific Open Database Connectivity data source with a client application.

ODBC driver manager

Provides the interface from the host language to the specific back-end data source driver.

operator access level

A number in the range 0 - 5 that designates the tasks the operator is authorized to perform. There must be at least one level 5 operator. This level has full permissions to customize the system, set up other operators, assign their permission levels, and so on.

-P-

password

Your password is a word or set of characters that confirms your identity. The password is never displayed or printed for anyone.

At your first login, enter the password that the Administrator assigned to you, then change your password using the SECURITY | PASSWORD menu. Otherwise, simply enter the password you selected at your first login.

A password can be up to 16 alphanumeric characters. MET/BASE ignores the case of the username and password. The password is NOT optional.

purge

To delete an asset from the database. Only level 4 and 5 operators can purge assets. When purged, the entire history of the asset is removed. Standards cannot be purged. Purging is done on the Inventory form.

-Q-

query

Commonly, a formalized instruction to a database to either return a set of records or perform a specified action on a set of records as specified in the query.

Also, the name of one of the MET/BASE applications, so called because it provides an interface for users to seek information from the database.

-R-

replication

For databases, a process by which the changes to data in one database (including creation, updating, and deletion of records) are also applied to the corresponding records in other databases.

RTAG

A special field in the database that serves to link tables to one another so that all the information about a particular maintenance event ("repair" event) can be retrieved.

The rtag links a maintenance event to the results generated during the event and to the standards used. The rtag is the unique identifier that the software assigns to each maintenance or repair event. Also see ctag and mtag.

reverse traceability

To trace those instruments that a subject instrument calibrated directly, or indirectly.

Also see traceability and forward traceability.

-S-

segregate

A subdivision of the database that permits restricting user access to a particular subset of data.

server

The term server is widely used to mean several different things.

Just as with the term "client", the word requires modification, otherwise it may be ambiguous whether it refers to a program or to the computer on which the program is running.

Whenever the word "server" is used without modification, it means the system designed to share data with client applications; servers and clients are often connected over a network. A server can simply share files or resources (such as printers), or it

can act as host for a database management system designed to provide logical access to a database. A database server usually contains and manages a central repository of data that remote client applications can retrieve and manipulate.

Standard (1) (Measurement Standard)

An object, artifact, instrument, system, or experiment that stores, embodies, or otherwise provides a physical quantity that serves as the basis for measurements of that quantity. In the context of MET/BASE, a standard is the next object in the traceability tree above the current level. For example, If we are calibrating a Fluke 87 (the UUT) in the RunTime, a standard could be a Fluke 5700.

See forward traceability.

Standard (2) (Paper Standard or Protocol)

A document that describes the operations and processes that must be performed in order for a particular end to be achieved. In Europe, the term "protocol" is preferred to avoid confusion with a physical (measurement) standard. For example, ISO Guide 25 describes the requirements for a calibration laboratory.

standard security

Uses SQL Server's own login validation process for all connections. To log into a SQL Server, each user must provide a valid login ID and password.

Stored Procedure (SP)

Stored procedures are programs kept in the database itself. They can be called from client applications. Stored procedures provide a way of providing uniform access to important functions automatically, as the procedure is held in the database, not in each client application.

Structured Query Language (SQL)

A database query and programming language.

SQL is commonly pronounced "sequel" and was originally developed for IBM mainframe computers. There is an ANSI-standard for SQL.

SQL Server

A relational database management system that is capable of supporting hundreds of simultaneous users, depending on your application and hardware environment.

System Administrator

The system administrator (SA) is the person responsible for the administrative and operational functions that are independent of any particular application, and is likely

to be a person with a comprehensive overview of the environment. Not to be confused with the Database Administrator.

System administration tasks typically include such things as installing software; configuring servers and clients; managing and monitoring the use of disk space, memory, and connections; creating devices and databases; authorizing users and granting them permissions; transferring data in and out of SQL Server databases; backing up and restoring databases; and diagnosing system problems.

The system administrator operates outside the protection system, which means that the Server does no permission checking for the system administrator. The system administrator is also treated as the owner of whatever database he or she is using. Anyone who knows the SA password can log in and act as system administrator.

Contrast with Database Administrator.

-T-

table

The basic unit of data storage in a relational database. A table stores data in records (rows) and fields (columns) and usually contains information on a particular category of things, such as employees or parts. Also called a base table.

TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is a network protocol supported by SQL Anywhere.

tool

One of the programs that make up the application. MET/CAL features the following tools: Runtime, Editor, Manual Entry, Database Setup, and Workstation. MET/TRACK has these tools: Manual Entry, Import Data, and Database Setup.

traceability

Traceability represents an unbroken chain of comparisons from the measurement being made, to a recognized, national, legal standard.

Also see forward traceability and reverse traceability.

trace report

A report that establishes an instrument’s traceability. MET/TRACK provides forward and reverse traceability for an asset to a selectable depth.

Also see forward traceability and reverse traceability.

transaction log

A log storing all changes made to a database, in the order in which they are made. In the event of a media failure on a database file, the transaction log is essential for database recovery. Therefore for optimal security, the transaction log should be kept on a different device from the database files.

transaction log mirror

An identical copy of the transaction log file, maintained at the same time. Every time a database change is written to the transaction log file, it is also written to the transaction log mirror file.

A mirror file should be kept on a separate device from the transaction log, so that if either device fails, the other copy of the log keeps the data safe for recovery.

TUR

Test Uncertainty Ratio, the specified uncertainty of the test instrument divided by the uncertainty of the calibrating instrument. A reliable TUR is obtained only when the specifications for the two instruments are correlated according to their respective uncertainty specifications and confidence intervals.

See Appendix D for details and example calculations.

-U-

user name

A user name is a name assigned to a login ID for the purpose of allowing a user to have access to a database. The abilities a user has within a database depend on the permissions granted to the user name.

User names are assigned by the Administrator. A user name is a unique, single word that may be as short as a single letter, but contains no more than 16 alphanumeric characters. The software enforces upper-case only user names.

-V-W-X-Y-Z-

uncertainty

An estimate of the possible error in a measurement. More precisely, an estimate of the range of values which contains the true value of a measured quantity. Uncertainty is usually reported in terms of the probability that the true value lies within a stated range of values.

validation

Validation restricts the data allowed in a field to a set of authorized, or valid, values. The software uses validation files to store the allowed values. These files are specified on a field-per-field basis as a part of customization; each field's validation file has a unique name, by default, the field number.

Index

—C—

—D—

—E—

—F—

—R—

—S—

—T—

—U—

—V—