

MINTEK

Specialists in mineral and
metallurgical technology

TECHNICAL MEMORANDUM

No 12741

STAFF ACTIVITY DISPLAY

by

Ashley Mulholland

RESTRICTED

17 January, 2005

**200 Hans Strijdom Drive
RANDBURG
South Africa**

This document is a means of internal communication within Mintek, and its distribution is restricted to those names appearing in the distribution list at the end. The Memorandum must not be cited as a reference or be included in any bibliography.

Measurement and Control

STAFF ACTIVITY DISPLAY

by

Ashley Mulholland

17 January, 2005

Chief Investigator: Ashley Mulholland

Supervisors Kobus Oosthuizen
 Chris Hockaday
 Stephen Kew

Head Paul Brereton-Stiles

Manager – Measurement and Control Dr D. G. Hulbert

Technical Objective No.: 610

Project No : 63320301

CONFIDENTIAL

SUMMARY

The purpose of the project was to provide the secretaries with an instantaneous view of the whereabouts of all Measurement and Control division staff members at all times, to ensure that client enquiries are directed to an appropriate, available staff member.

The project involved the development of three separate module's communicating via a remote connection. The first was a client application running on each staff member's computer, allowing for input of staff member whereabouts. The second was the central server application managing all staff activities and regularly backing this data up in case of system failure. The third was a client on one or more of the secretaries' computers displaying all the staff member activities.

CONFIDENTIAL

TABLE OF CONTENTS

- 1. INTRODUCTION 1
- 2. EXPERIMENTAL PROCEDURES 2
 - 2.1. Design Process 2
 - 2.2. Remote Communication 3
 - 2.2.1. *Data Flow Between Applications* 3
 - 2.2.2. *Objects used in Remote Communication* 3
 - 2.2.3. *Configuration Files* 4
- 3. APPLICATION FUNCTIONALITY 5
 - 3.1. Features 5
 - 3.1.1. *Staff Client Application* 5
 - 3.1.2. *Server Application* 5
 - 3.1.3. *Display Client Application* 5
- 4. ADDITIONAL INFORMATION 7
 - 4.1. Method of Operation 7
 - 4.1.1. *Installation and Start-up* 7
 - 4.1.2. *Setting Up a Staff Member Account* 7
 - 4.1.3. *Operation of Staff Client Application* 8
 - 4.1.4. *Operation of Display Client Application* 8
 - 4.2. Future Improvements 9
- 5. CONCLUSION 10
- 6. REFERENCES 11
- 7. APPENDIX 12
 - 7.1. Project Specifications 12

CONFIDENTIAL

LIST OF FIGURES

Figure 1. Data Flow between Applications..... 3

Figure 2. Staff Details Interface..... 7

Figure 3. Staff Client Application Interface 8

Figure 4. Display Client Application Interface..... 8

CONFIDENTIAL

1. INTRODUCTION

Client enquiries need to be directed to an appropriate and available staff member. This means that the secretary needs to know the location of all staff member's at all times.

Currently, the Measurement and Control division of Mintek makes use of a white-board in the secretary's office on which staff members can indicate any trips away from Mintek during the day. This has its limitations. Not only is it inconvenient, but it also does not account for times during the day when staff members, even though they are at Mintek, are unavailable (for example, when they are busy in meetings). Although meetings and other activities are indicated on calendars in Microsoft Outlook, the secretary still does not have an instantaneous view of the staff member whereabouts.

The aim of the project was to develop an instantaneous, reliable and accurate mechanism, allowing the secretary to view the location of any staff member at a glance.

In order to achieve this three separate programs were written. They communicate with each other via a remote connection via the existing Ethernet network to which all computers are linked. The Staff Client application is installed on each staff member's computer, allowing for input of staff member whereabouts. The Server application, installable on any computer on the network, manages all staff activities and regularly backs this data up in case of system failure. The Display Client application is installed on one or more of the secretaries' computers, allowing for the display of all staff member whereabouts.

The project was developed using Microsoft Visual Studio Dot Net 2003, in which C# was used. Remote communication was achieved using DotNet Remoting.

CONFIDENTIAL

2. EXPERIMENTAL PROCEDURES

2.1. Design Process

The project began with research into the topic of DotNet Remoting. Where ever necessary sketches were made to understand the problem and aid in solving it. Experimentation was necessary throughout the project, so as to obtain a working solution. The following steps were taken:

- Initially, a simplistic version of the project was written in order to achieve remote communication between two applications (namely one staff client application and the server application).
- Once this was achieved a simplistic display client application (without the Developer Express XtraGrid interface) was written.
- The server application was upgraded (a staff member collection was added) to allow for many staff clients.
- After this the interface of the display client was upgraded to the XtraGrid format using Developer Express Inc.NET.
- Data storage and retrieval were implemented on the staff client application and the application used to obtain staff member details (name and group) was written and integrated into the staff client application.
- Error handling was incorporated in the staff client and display client application, in case of server failure.
- Data storage and retrieval were implemented on the server application.
- All three of the applications were then converted to windows services, but it was found that services, by default, have no interface, and so both the staff client application and the display client application were written as applications that would have shortcuts installed to the startup folder of the computer.
- Setup projects were written for all three applications, allowing them to be installed with ease.
- Finally, adjustments were made to configuration file data and remoting procedures in the source code. The functionality of the staff client and display client applications, while the server is unavailable and when it becomes available again, was also added.

CONFIDENTIAL

2.2. Remote Communication

2.2.1. Data Flow Between Applications

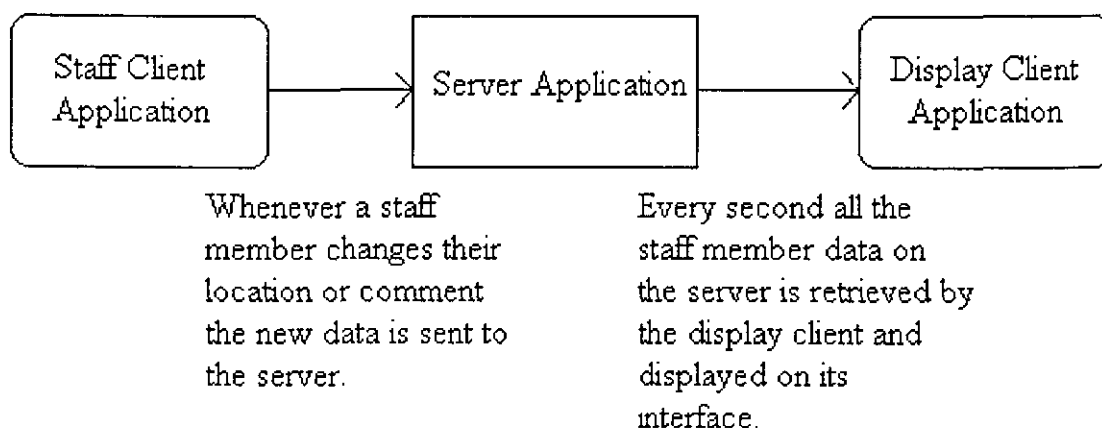


Figure 1. Data Flow between Applications

The three separate modules seen in Figure 1 communicate via a remote connection. This remote communication was achieved using DotNet Remoting.

2.2.2. Objects used in Remote Communication

Remote communication makes use of two kinds of objects, those that are passed by reference and those that are passed by value.

MarshalByRefObjects are remote objects that run on the server and allow remote method calls from the client to be executed on the server-side. These objects are found only on the server and only a so-called ObjRef will be passed around. The ObjRef is like a networked pointer which shows on which server the object exists (contains the server name/IP address) and contains the object identity, identifying exactly one object on the server. The client does not have the compiled object in any of its assemblies, and so source code needs to be generated using SoapSuds so that the client will know the server's definition, the required SoapMethodAttributes and hence the server's namespace identifier. These C# files are generated using the WSDL information provided by the server. To generate a code file the following line is entered into the Visual Studio .NET 2003 Command Prompt while the server is running:

```
soapsuds -url:http://localhost:1234/StaffActivity.soap?wsdl -nowp -gc.
```

The generated C# source code file will be placed in the current directory. This file needs to be included and referenced in the client-side project. When using MarshalByRefObjects every method is executed on the server. When a creation request is encountered on the client (for example, the line of code: `StaffActivity stf = new StaffActivity()`), an activation message is sent to the server, where a remote object is created. On the client a proxy is created that holds the ObjRef to the server object. The proxy object looks just like a local object to the client. Methods can then be executed on the server via the proxy object on the Client.

CONFIDENTIAL

The second kind of object is the ByValue object. These objects, when passed over remoting boundaries (as method parameters or return values), are serialized into a string or binary representation. They are then stored as a copy on the other side of the communications channel. This means that after the ByValue object has been passed, both the client and the server each have their own copy, and so run independently of each other. Methods called on these objects are executed locally, by the application that called the method. In other words, a method called by the client will be executed on the client and a method called by the server will be executed on the server. Therefore, methods are not limited to being executed only on the server as with MarshalByRefObjects. This means, though, that the compiled object needs to be contained in the client and server assemblies. This substantiates the creation of a general class library that contains all ByValue object classes and their methods. Both the server and the client then reference this general class library, so that all ByValue object classes are available to each. Another requirement for a ByValue object is that it supports serialization, which allows the object to be passable as a string or a binary representation. This can be achieved by marking the ByValue object class in the general class library with the class level attribute: [Serializable].

Both MarshalByRefObjects and ByValue Objects were used in this project to allow remote communication between the three different modules.

2.2.3. Configuration Files

Configuration files were used to specify the remoting parameters, allowing the remoting parameters to be altered at any time, even after installation, without having to alter the source code. Remoting parameters in the server configuration file include: the reference for a predefined channel ("tcp" or "http"), the server side port number (For example, "1234") and whether you want to provide remote access to a Singleton or SingleCall object on the server. With SingleCall objects an object is created by the server, a method is executed on it and then it is destroyed. When wanting to share data or resources between clients, as in this project, Singleton objects should be used as they have a longer lifetime (they are destroyed after a few minutes by default).

Remoting parameters in the client configuration file include: the full URL to the server's registered object and the type information in the form "<ServerNamespace>.<ServerClassname>, <ClientAssembly>". The URL varies according to the name of the server computer, and so it must be ensured after installation that the URL in each client configuration file contains the server computer's name. For more information on this see section 4.1.1.

To employ a configuration file in an application RemotingConfiguration.Configure() is called and the filename of the configuration file is passed to it (for example, RemotingConfiguration.Configure("client.exe.config")).

CONFIDENTIAL

3. APPLICATION FUNCTIONALITY

3.1. Features

3.1.1. *Staff Client Application*

The staff client application runs on each staff member's computer. It is installed so as to run automatically when the computer is started, in order to ensure that staff members remember to use it. The interface consists of a combo box, which allows the staff member to select a location from the options "In", "Out" and "Meeting". A textbox is also included where the staff member can enter optional comments. Since the application runs continuously on each staff member's computer it is easy to use and non intrusive or distracting. When the program is first run it asks the user for their name and group (either Furnace, Milling or Flotation). These details, along with the staff member's current location and comment, are then displayed in the Display Client Application on the secretary's computer.

Whenever the staff member changes the location or comment the new data is stored in text files on the client computer. Therefore, when the staff client application is terminated and then restarted, the stored data is read from the files and the last information entered on the interface will be displayed. The server and display client are unaffected when the staff client application is terminated.

If the server is unavailable the staff member's location and comment can still be entered, but the changes to their whereabouts are not updated to the display client on the secretary's computer. A timer in the staff client application checks the availability of the server and when the server becomes available again the changes to the staff member whereabouts are automatically updated to the display client.

3.1.2. *Server Application*

The server application can be installed on any computer on the network. All clients (both staff member and secretary clients) connect to it. It is run as a service, ensuring that it runs automatically on start-up. This is essential as the server must, as far as possible, always be accessible to the clients.

Current data is stored on the server in a collection of staff member objects. Every second, and on closing, the collection data is stored to a data file on the server computer. This stored data is loaded into the collection on the server when the server is started. Therefore, if the server computer is shutdown or crashes, the staff member data, including each staff member's name, group, location and comment, is loaded back into the server collection when the server is restarted. This ensures that no staff client information is lost and that it is made readily available to the display client immediately after the server has been restarted.

3.1.3. *Display Client Application*

The display client application shows all staff members and their whereabouts. Staff members' names are grouped according to their group (Furnace, Milling or Flotation) and then arranged alphabetically. This grouping and alphabetical arrangement can be altered by the user. The

CONFIDENTIAL

display client application can be placed on any number of computers (though it will normally only be placed on the secretaries' computers).

If the server is unavailable the client display application will still continue to run, but any changes made to the staff members' whereabouts will not be updated to it until the server becomes available again. If the display client application is terminated, all other clients and the server are unaffected. When the display client application is restarted all the current data on the server will be retrieved by it.

CONFIDENTIAL

4. ADDITIONAL INFORMATION

4.1. Method of Operation

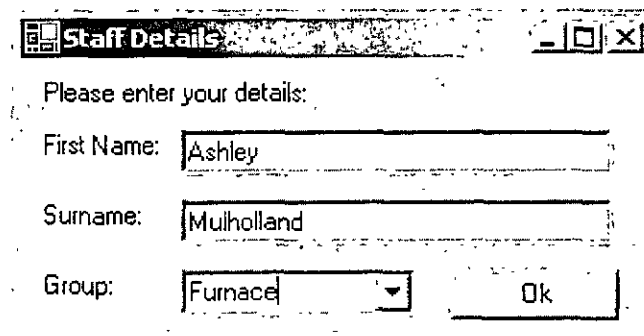
4.1.1. Installation and Start-up

All three applications first need to be installed on appropriate computers - the staff client application on the staff computers, the display client application on the secretaries' computers, and the server application on any reliable computer (probably on one of the secretaries' computers). This is easily done by just running the setup of each of these applications. All that needs to be done after this is to ensure that all the configuration files of the clients contain the server computer's name in the following line of code:

```
url="tcp://ServerComputerName:1234/StaffActivity.soap"
```

All the applications run on start-up, but shortcuts are contained in the Programs menu for the staff client application and the display client application for convenience. The server application is a service and has no interface. Therefore, the server computer needs to be restarted after the server application is installed in order for the server application to run.

4.1.2. Setting Up a Staff Member Account



The screenshot shows a window titled "Staff Details" with a close button. The text "Please enter your details:" is at the top. Below it are three input fields: "First Name:" with the text "Ashley", "Surname:" with the text "Mulholland", and "Group:" with a dropdown menu showing "Furnace". An "Ok" button is located to the right of the Group dropdown.

Figure 2. Staff Details Interface

When the staff client application is first run the staff details interface (see Figure 2) will be displayed. This allows the staff member to enter his or her name and group (Furnace, Milling or Flotation). The server must be available during this time, as the user's account needs to be set up on the server. If the server is not available an error message will be raised and the program will terminate. The next time the staff client application is run the staff details interface will be displayed again, but, for convenience, it will already contain the previously entered information. After the staff details have been successfully completed the staff client application interface will be displayed.

CONFIDENTIAL

4.1.3. Operation of Staff Client Application

Figure 3. Staff Client Application Interface

Figure 3 shows the interface of the staff client application. The staff member's name and group are displayed in the caption, except when the server is unavailable and the caption is set to display "Server Unavailable". This is so that the user always knows the availability of the server and, hence, whether the changes to their whereabouts have been updated to the display client. The staff member can select between the three location options, "In", "Out" and "Meeting". An optional comment can also be entered. Sometimes it may be useful for the comment to contain a return time, and so a reminder of this was placed in the comment label.

4.1.4. Operation of Display Client Application

Group	First Name	Surname	Location	Comment/Return Time
	John	Smith	Out	Shopping/15.00
	Joshua	Peace	In	
	Susan	Blakey	Out	Lunch with Jane
	Ashley	Mulholland	In	Working
	Mary	Walker	Meeting	None
	Scott	Brown	Meeting	with Don/11.00
	Kobus	Oosthuizen	In	Working very hard again....
	Paul	Hugo	Out	See you later

Figure 4. Display Client Application Interface

Figure 4 shows the interface of the display client application. All the staff members and their whereabouts are listed here. In the default case, staff members are grouped together according to their group (Furnace, Milling or Flotation). They are then arranged alphabetically, according to their first names, within these groups. However, this arrangement can be altered. Staff members can be grouped according to any column heading by dragging the column heading into the grouping area (the area where "Group" is located in Figure 4) and then dragging the "Group" heading from the group area back to be a column heading. For example, it may be useful to group staff according to their Location. It may also be useful to have no grouping at all by just dragging the "Group" heading back to be a column heading. The staff members can be arranged alphabetically (ascending or descending) according to any

CONFIDENTIAL

of the column headings by simply clicking on the column heading. For example, it may be useful to *arrange the staff alphabetically according to their surname*. When the server is unavailable the caption at the top will display "Server Unavailable". This is so that the user will always know the availability of the server and hence the accuracy of the information displayed.

4.2. Future Improvements

There are possibilities for future improvement of the staff activity display. Firstly, the server application could interact with Microsoft Outlook's calendars, to extract staff member activities directly from their calendars. Also a board could be erected in the secretary's office with lights to indicate everyone's whereabouts. This would allow outsiders, as well as staff, quick access to the availability of any staff member.

CONFIDENTIAL

5. CONCLUSION

Remote communication between the clients and the server means that secretaries have real-time access to the whereabouts of any staff member. The regular backing-up of staff member data on the server ensures that data is not lost if the server or any of the client computers crash or shut down due to a power failure, for example. Each client, although dependent on the server for communication, still runs in the absence of the server and automatically updates when the server becomes available again. User-friendly installation processes and interfaces ensure ease of use.

These features ensure that the staff activity display offers an instantaneous, reliable and accurate mechanism, allowing the secretaries to view the location of any staff member at a glance.

6. REFERENCES

Ingo Rammer: "Advanced .NET Remoting", Impressions Book and Journal Services, Inc., 2002.

7. APPENDIX

7.1. Project Specifications

MINTEK

Measurement and Control Division

Vacation Student Project 2004 / 2005

Student: Ashley Molholland

Supervisors: Kobus Oosthuizen

Chris Hockaday

Stephen Kew

Staff Activity Display

Introduction

To ensure that client enquiries are directed to an appropriate staff member who is available to handle the enquiry, it is important for a secretary to know the whereabouts of all Measurement and Control division staff members at all times.

Currently a white-board in the secretary's office is used to indicate staff trips away from Mintek. It however does not provide any information on staff activities during a typical day at Mintek, e.g. being in a meeting and hence temporarily not being available to handle client enquiries. Meetings and other activities are indicated on calendars in Microsoft Outlook, but it still does not provide the secretary with an instantaneous view of the availability of all staff members. A reliable and accurate mechanism is thus required to map all staff activities to a central location, from where a secretary can view it in a single glance.

Scope

The project involves the development of three separate modules, communicating via a remote connection. The first is an application running on each staff member's computer, the second

CONFIDENTIAL

is the central server application managing all staff activities and the third is a client displaying all the client activities on one or more secretaries' computers.

The project should be developed using Microsoft Visual Studio Dot Net 2003, preferably using C#. Remote communication should be done using DotNet Remoting.

Staff Client application

The client application would run on each staff-member's computer, and need to be running as a service that would start-up automatically when the computer is started. The staff members need to indicate their whereabouts by selecting from a predefined number of options (e.g. "In", "Out", "Meeting"). A comment section also needs to be included to which the user should optionally add comments. As this application would be running continuously on each staff member's computer, it needs to be ensured that the application is easy to use and not intrusive or distracting. This application should also contain a configuration tool (to be used when installing the client) to configure the client's display name and group in the Measurement and Control division.

Server Application

The server application would be installed on a computer in the secretary's office, to which all other staff members should connect. This application should also run as a service. This application should manage all staff activities, and allow connections from staff computers as well as another client installed on secretaries' computers displaying all the staff's whereabouts.

Display Client

A display module should be developed to show all staff members and their whereabouts. Staff members' names should be grouped by Group (Furnace, Milling and Flotation) and then alphabetically. Next to each staff member's name, the staff member's whereabouts should be indicated (using e.g. a radio button to select one of the 3 options) and also the comments if any.

Issues to consider

- Maintaining states when a client computer shuts down.
- Client connection to the server computer.
- Client reconnection after being disconnected (e.g. server computer shutting down).

CONFIDENTIAL

- Client updates when no server is present.
- Configuration of the server computer to connect to.
- User configuration required to identify user.
- Ordering of users

Documentation:

Code should be commented thoroughly and all actions be documented, to allow future upgrades on this project. MaC coding standards should be followed.

Presentation and Report

A few days before you finish, you should be prepared to give a 10-minute presentation of your work to a small group of people. The presentation should be Powerpoint based. This presentation should include a demonstration of the project.

You should also prepare a formal report in the form of a Technical Memorandum, with this document given as an appendix. Check with your supervisor on what you can use to report on back to the university. Generally, your report will be the TM with a different cover, but sometimes sensitive material needs to be omitted or disguised.

Additional functionality

A number of additional features are envisaged to improve the project in future. The first is to have the server module interacting with Microsoft Outlook's calendars, to extract staff activities directly from their calendars. Should the project be completed before the due date, the student should start exploring this functionality. This is however optional functionality, and fully functional version excluding this functionality, as well as completed documentation is a prerequisite for this to be explored.

CONFIDENTIAL

DISTRIBUTION

DG Hulbert

B Tait

K Oosthuizen

C Hockaday

S Kew

A Mulholland

File 63320301

