

# **An Introduction to Microsoft Word 2007:**

## **Citations & Bibliography**

---

### **Introduction:**

In any academic document it is vital to reference the sources that have been used in the main body of the text. A vast number of references styles exist for the multitude of research disciplines. This document aims to show how to use the Citation and Reference features available in Microsoft Word.

### **Creating the text to reference:**

Create a new Word document and enter the text as shown below:

**ABSTRACT**

*In this paper we describe the design and implementation of a Windows service based software agent toolset to classify student programmers. The graphical element of the toolset presents lecturers with the complexity level at which a student programmer is operating at. With an accurate complexity level it is possible to place appropriate learning resources to challenge the more advanced student and also to support the less advanced.*

**Keywords**

*Automatic Classification, .NET Windows Services, Software Agent*

**1. Introduction**

To make student programmers more productive and technically competent requires both education and experience. To identify the training requirement necessitates detailed knowledge of the students' skill set and the program code they currently use in order to generate complexity data. Over time, student programmers will create and develop many software solutions. Generating the programming complexity data from these solutions requires a significant amount of storage that ideally is saved within a relational database environment. The saved data is rich with hidden information, and with the right transformations and tools, can be used to classify student programmers. Thus, the people who most require support can be identified. There are several classification schemes reported in the literature. The classification process carried out by Govindarajulu involved the use of on-line questionnaires. This process is superior to that of paper-based questionnaires in that the data received is in a machine readable format and immediately available for analysis. The process takes time out of the end users normal schedule and questions can be misinterpreted. A superior approach is to gather the data automatically via software agent based toolkit. This creates an accurate picture of the end user and with data gathered on a continuous basis enables many snap-shots of the data to be captured. Multiple snap-shots over time gives the opportunity to measure the progress a student programmer has achieved. It is proposed to create a set of levels that programmers can work towards to gain a classification of their work. These classifications will be arrived at by analysing the user's source code over a period of time and checking that code for features for each level of classification.

**2. Agent Based Windows Service**

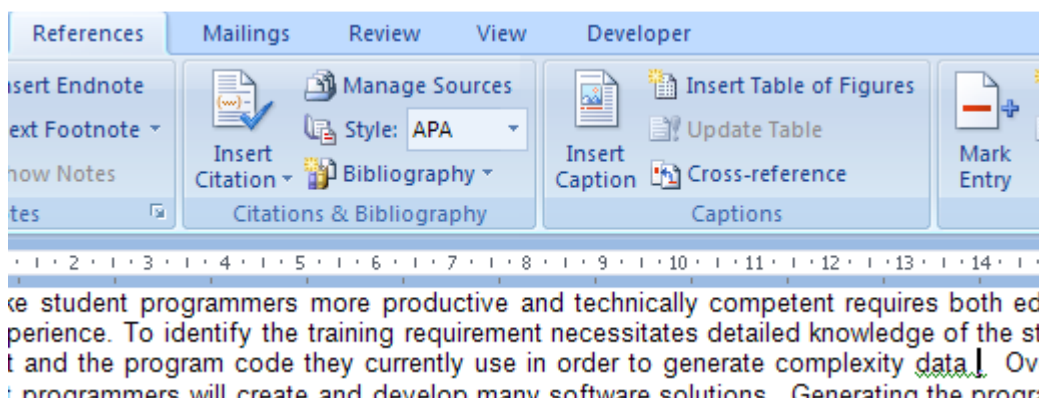
An agent based on the concept of a Windows service has been explored by [5-9]. Windows services enable developers to create long-running executable processes that run an application in the background. The services can be automatically started when the operating system loads into memory, but can also be paused and restarted manually, either using the Microsoft Management Console (MMC) utility, or a third party utility. The authors have used this concept in the construction of the data gathering system that is configured to start automatically at user log-on to continuously monitor and analyse the code generated by student programmers.

A Windows Service is designed to not require user intervention and to run unattended and autonomously. To run a Windows service it must first be installed on the host machine and then the service will be started as a process when the computer boots up. Visual Studio ships components to install service applications as this type of application by itself is not capable of being installed.

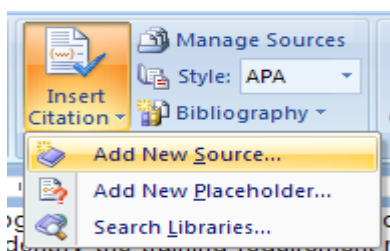
After installers are added to the application, the next step is to create a setup project that will install the compiled project files and run the installers needed to install the service. In order to create a complete setup project, the service project's output must be added to the setup project and then a custom action must be added to have the service installed.

The second sentence in the introduction needs to be referenced to its source to meet with academic convention.

1. Place the cursor immediately before the full stop of the second sentence leaving a single space after the word *data*.
2. Select the **References** tab.



3. From the Citations and Bibliography section, click the  button.
4. Next click the **Add New Source** option.

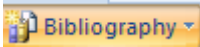


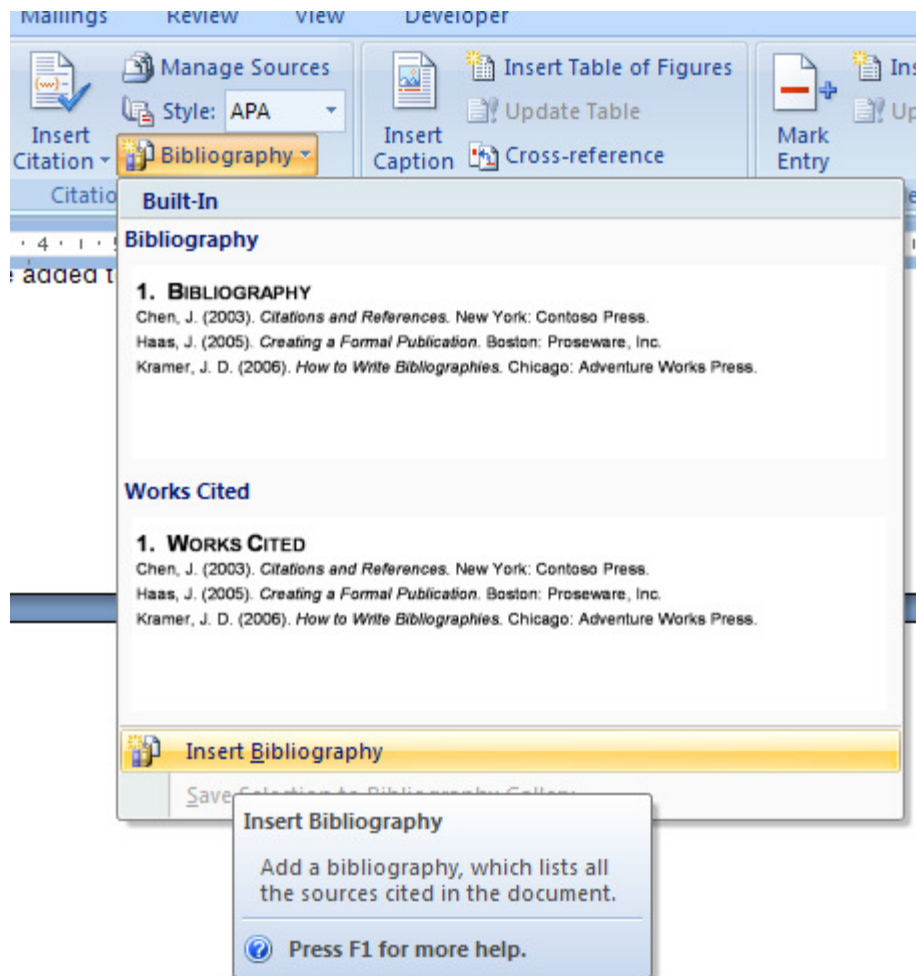
5. On the Create Source form fill in the details of the publication as shown below:

6. The document will now show in the text the appropriate reference.

## 1. INTRODUCTION

To make student programmers more productive and technically competent requires both education and experience. To identify the training requirement necessitates detailed knowledge of the students' skill set and the program code they currently use in order to generate complexity data (Hole, McPhee, & Mhereeg, 2008).

7. At the end of the document create a page break.
8. At the top of the final page add the heading **References**.
9. Select the References tab.
10. From the Citation and Bibliography section click on the  button and Insert Bibliography.



11. The Reference section of the document now appears as below:

### 3. REFERENCES

Hole, S., McPhee, D., & Mhereeg, M. (2008). Developing a Windows Agent to Analyse Spreadsheet and Macro Function Complexity. *4th International Conference on Computer Science and Information Systems* (pp. 28-36). Athens, Greece: ATINER Publications.

12. The text below shows the references required in the academic paper and points 13-17 show the details of the source articles

### Introduction

To make student programmers more productive and technically competent requires both education and experience. To identify the training requirement necessitates detailed knowledge of the students' skill set and the program code they currently use in order to generate complexity data (Hole, McPhee, & Mhereeg, 2008).

Over time, student programmers will create and develop many software solutions. Generating the programming complexity data from these solutions requires a significant amount of storage that ideally is saved within a relational database environment. The saved data is rich with hidden information, and with the right transformations and tools, can be used to classify student programmers. Thus, the people who most require support can be identified. There are several classification schemes reported in the literature (Rockart & Flannery, 1983) (Cotterman & Kumar, 1989). The classification process carried out by Govindarajulu involved the use of on-line questionnaires (Govindarajulu, 2003). This process is superior to that of paper-based questionnaires in that the data received is in a machine readable format and immediately available for analysis. The process takes time out of the end users normal schedule and questions can be misinterpreted. A superior approach is to gather the data automatically via software agent based toolkit. This creates an accurate picture of the end user and with data gathered on a continuous basis enables many snap-shots of the data to be captured. Multiple snap-shots over time gives the opportunity to measure the progress a student programmer has achieved. It is proposed to create a set of levels that programmers can work towards to gain a classification of their work. These classifications will be arrived at by analysing the user's source code over a period of time and checking that code for features for each level of classification.

### Agent Based Windows Service

An agent based on the concept of a Windows service has been explored by (Hole & McPhee, Constructing an Agent to Facilitate Automatic Collection of End-User Data, 2006) (Hole & McPhee, Constructing an Agent to Facilitate the Automatic Collection of End-User Desk-Top Data, 2006). Windows services enable developers to create long-running executable processes that run an

13. Reference 2 is show below:

The 'Create Source' dialog box is shown with the following fields and values:

- Type of Source: Journal Article
- Language: Default
- Bibliography Fields for APA:
  - Author: Rockart, J; Flannery, S
  - Corporate Author: (unchecked)
  - Title: The Management of End-User Computing
  - Journal Name: Communications of the ACM
  - Year: 1983
  - Pages: 776-784
- Show All Bibliography Fields: (unchecked)
- Tag name: Roc83
- Example: 50-62
- Buttons: OK, Cancel

14. Reference 3 is shown below:

The 'Create Source' dialog box is shown with the following fields and values:

- Type of Source: Journal Article
- Language: Default
- Bibliography Fields for APA:
  - Author: Cotterman, W; Kumar, K
  - Corporate Author: (unchecked)
  - Title: User Cube: A Taxonomy of End-Users
  - Journal Name: Communications of the ACM
  - Year: 1989
  - Pages: 1313-1320
- Show All Bibliography Fields: (unchecked)
- Tag name: Cot89
- Example: 50-62
- Buttons: OK, Cancel

15. Reference 4 is shown below:

The 'Create Source' dialog box is shown with the following fields and values:

- Type of Source: Journal Article
- Language: Default
- Bibliography Fields for APA:
  - Author: Govindarajulu, C
  - Corporate Author: (unchecked)
  - Title: End-Users Who are They?
  - Journal Name: Communications of the ACM
  - Year: 2003
  - Pages: 152-159
- Show All Bibliography Fields: (unchecked)
- Tag name: Gov03
- Example: 50-62
- Buttons: OK, Cancel

16. Reference 5 is shown below:

**Create Source**

Type of Source: Book Section Language: Default

Bibliography Fields for APA

Author: Hole, Stephen; McPhee, Duncan [Edit]  
☐ Corporate Author

Title: Constructing an Agent to Facilitate Automatic Collection of End-User Data

Book Author: Petratos, P [Edit]

Book Title: Current Computing Developments in E-Commerce, Security and Collaborative Systems

Year: 2006

Pages: 41-60

City: Athens, Greece

Publisher: ATINER Publications

☐ Show All Bibliography Fields

Tag name: Example: Adventure Works Press  
 Hol06

OK Cancel

17. Reference 6 is shown below:

**Create Source**

Type of Source: Conference Proceedings Language: Default

Bibliography Fields for APA

Author: Hole, Stephen; McPhee, Duncan [Edit]  
☐ Corporate Author

Title: Constructing an Agent to Facilitate the Automatic Collection of End-User Desk-Top Data

Pages: 197-208

Year: 2006

Conference Publication Name: 3rd International Conference on Telecommunications and Computer Networks

City: Portsmouth, UK

Publisher: IADAT-tcn Publications

☐ Show All Bibliography Fields

Tag name: Example: Adventure Works Press  
 Hol061

OK Cancel

18. To update the Reference section of the document Right click on the existing entry in the bibliography and then select Update. The result is shown below:

### 3. REFERENCES

Cotterman, W., & Kumar, K. (1989). User Cube: A Taxonomy of End-Users. *Communications of the ACM*, 1313-1320.

Govindarajulu, C. (2003). End-Users Who are They? *Communications of the ACM*, 152-159.

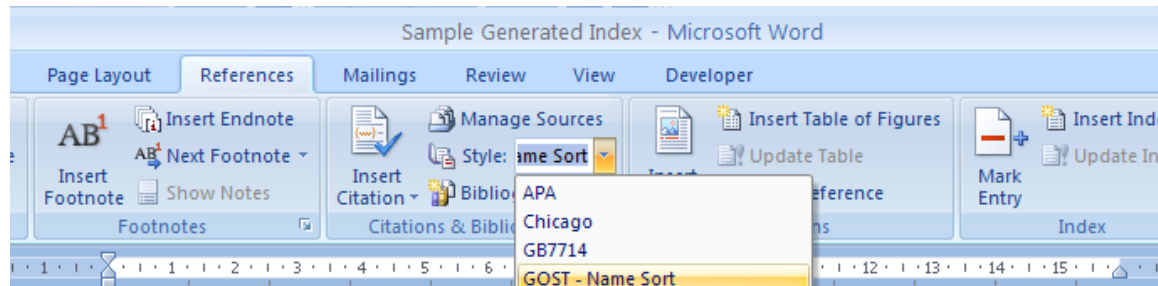
Hole, S., & McPhee, D. (2006). Constructing an Agent to Facilitate Automatic Collection of End-User Data. In P. Petratos, *Current Computing Developments in E-Commerce, Security and Collaborative Systems* (pp. 41-60). Athens, Greece: ATINER Publications.

Hole, S., & McPhee, D. (2006). Constructing an Agent to Facilitate the Automatic Collection of End-User Desk-Top Data. *3rd International Conference on Telecommunications and Computer Networks* (pp. 197-208). Portsmouth, UK: IADAT-tcn Publications.

Hole, S., McPhee, D., & Mhereeg, M. (2008). Developing a Windows Agent to Analyse Spreadsheet and Macro Function Complexity. *4th International Conference on Computer Science and Information Systems* (pp. 28-36). Athens, Greece: ATINER Publications.

Rockart, J., & Flannery, S. (1983). The Management of End-User Computing. *Communications of the ACM*, 776-784.

19. On the Citations and Bibliography section of the References tab it is possible to change the style of the references.



## ABSTRACT

*In this paper we describe the design and implementation of a service based software agent toolset to classify student programmers. The graphical element of the toolset presents lecturers with the complexity level at which a student programmer is operating at. With an accurate complexity level it is possible to place appropriate learning resources to challenge the more advanced student and also to support the less advanced.*

## Keywords

*Automatic Classification, .NET Windows Services, Software Agent*

## 1. INTRODUCTION

To make student programmers more productive and technically competent requires both education and experience. To identify the training requirement necessitates detailed knowledge of the students' skill set and the program code they currently use in order to generate complexity data (Hole, et al., 2008).

Over time, student programmers will create and develop many software solutions. Generating the programming complexity data from these solutions requires a significant amount of storage that

