

AUTOPAYITEM - A SOFTWARE TOOL TO MANAGE PAY ITEMS FOR HIGHWAY DESIGN

Michael Kyte
NIATT
University of Idaho
Moscow, ID 83844
mkyte@uidaho.edu
(208)885-6002

Zhuxin Yang
Texas Department of Transportation
1300 N. Texas Avenue
Bryan, TX 77803
zyang@dot.state.tx.us
(979)778-9657

Zong Tian
Texas Transportation Institute
The Texas A&M University System
College Station, TX 77843-3135
z-tian@tamu.edu
(979)862-2458

Wade Christiansen
Idaho Transportation Department
Boise, ID 83707-2028
WChristi@itd.state.id.us
208-334-8969

Text: 3485
Total Figures: 8
Total Tables: 2
Total Words: 5985

Submitted for Presentation and Publication at the 83rd TRB Annual Meeting
15 November 2003

Abstract

This paper documents the development and testing of AutoPayItem (API), a computer software tool that manages highway design pay items based on the current standards and procedures of the Idaho Transportation Department (ITD). API can be described as a MicroStation-based CADD tool that automatically keeps track of pay items as project plans develop, eliminating the shortcomings of the current manual procedures where the process is time consuming and error prone due to multiple entry and frequent modification of the same pay item information. The MicroStation Development Language (MDL) is the primary tool for coding the software. API allows highway designers a single point of entry for all pay item information through the use of a database, such as Oracle or Microsoft Access. All reports, including graphical summaries of the project plans, are derived directly from the data entered. Preliminary testing of the software by highway engineers at ITD indicated that the efficiency of pay item processing improved by more than 100 percent over the original manual method. API resulted in no errors compared to an average of 16 errors by the manual method. Surveys of pay-item procedures used by the Texas Department of Transportation and the New Jersey Department of Transportation showed similarities to the procedures used at ITD, although some differences exist in terms of the format for plan sheets, summaries and pay items. Therefore, with minor modifications, API presents a good potential for use by other state agencies to reduce their costs of preparing pay items during the highway design process.

Key Words: Highway Design, Pay Item, MDL

1. Introduction

The highway design process involves multiple tasks and steps. Plans, specifications, and estimates are the bridge between the design concept and the final product. Plans, specifications, and the bid process are part of the total contract that is made between a public agency and the private contractor who constructs a project. The quality of the plans is critical to the success of a highway design project. It is the designer's responsibility to prepare plans that are complete, clear, and correct.

The pay item is one of the most important elements of a highway design plan. A pay item is a specific unit of work for which a price is provided and paid to a contractor while a highway is under construction. The three major properties of a pay item include the item number, the specification, and the cost estimate. The item number is given uniquely on a design plan. The specification describes the material to be used, the method of incorporating the material, and how the completed work will be measured and paid for.

The cost estimate is made to ensure that the overall design falls within the budget available for the project. Incorporating pay item information correctly into design plans including plan sheets, summaries, the engineer's estimate and bid documents through the use of a computer-aided design and drafting (CADD) tool is a major task for highway designers. Mistakes in entering pay item information and errors due to change orders and pay item quantity increase the number of disputes with contractors, and often significantly increase the cost of a project. Highway designers have been looking for alternative tools and solutions to minimize pay item related errors.

During the highway design process, designers such as those at the Idaho Transportation Department (ITD) manually enter pay item related information into several different

locations, including the plan sheets, the sidebar summary, the roadway/bridge summary, and the engineer's estimate. Such information includes the pay item number, the name of the pay item, its quantities and stations. Designers must sort pay items according to the pay item number, calculate the total quantities, and enter this information at the correct location on the design plans. Designers must manually keep track of all pay items associated with a design as plan sheets, summaries and bid documents are developed, modified, and finalized.

The obvious drawbacks to the manual process are the time that it requires and its sensitivity to errors, since the same pay item information must be noted or entered in several locations at different times. Any revision to a specific pay item requires the designer to modify information stored in several locations in the highway design plans. The majority of pay item related errors are human errors, resulting from keyboard entry mistakes and failure to make changes in all applicable locations during the development of project plans and bid documents (1). Complex plans and drawings make the identification of such errors even more difficult. Correcting and checking pay item errors requires a significant amount of work.

Clearly, pay item errors should be kept to a minimum during the design process.

Highway designers at ITD feel an urgent need to automate the pay item management process in order to reduce or eliminate disputes with contractors due to pay item related errors. It is highly desirable to develop a software tool to automate the pay item management process at ITD. Over the past several years, researchers at the National Institute for Advanced Transportation Technology (NIATT) at the University of Idaho have been developing such a software tool, namely AutoPayItem or API. This paper

documents the development process of this tool and some of the initial testing results from the application of this tool. The overall process of pay item management at ITD and other states are discussed in section 2. The development process of API is documented in section 3. The main features of API and the initial testing results are presented in section 4. Finally, the summary and conclusions from this work are provided in section 5.

2. Pay Item Management Process at ITD and Other States

As part of this research, a survey was conducted on the procedures used by state DOTs for managing pay items in the highway design process. The states surveyed include the Idaho Transportation Department (ITD), the Texas Department of Transportation (TxDOT), and the New Jersey Department of Transportation (NJDOT). Although there are subtle differences among these states on how pay items are managed, they follow some basic principles and procedures. We first present the procedures used at ITD, and then provide information on the procedures used at TxDOT and NJDOT.

Management of pay items at ITD follows the procedures described below:

1. Designers select and design pay items on paper according to the ITD project workflow from the ITD Design Manual (2). They draw the plan sheet inside MicroStation using the MicroStation drafting tools. A sample plan sheet is shown in Figure 1.
2. Designers identify the location of the pay item on the plan sheet by drawing a capsule with or without a leader line using MicroStation drafting tools. A capsule is an oval shape that is designated for the pay item callout.

3. Designers check against an established master pay item list, find the specific pay item information and enter the pay item number inside the capsule oval. The same process is repeated until all pay items related to a particular project are entered. A sample plan sheet with pay item capsules is shown in Figure 2.
4. Designers manually enter pay item information on the right side of the plan sheet to create the sidebar summary. This information includes the pay item number with an oval, the pay item name, quantities, units, stations etc. A sample sidebar summary is shown in Figure 3. Designers complete all sidebar summaries following the same procedure.
5. The roadway/bridge summary, a project level summary, is created inside MicroStation. Designers manually enter the pay item master key number, the pay item name, the unit, the quantities and other pay item related information on this summary. Designers need to move back and forth between all plan sheets over and over again to enter and transfer all relevant information. The pay items are sorted by pay item number. Pay items that are plan sheet specific are sorted per plan sheet. All pay item quantity totals are calculated and manually entered. Designers break the pay items into multiple summary sheets as needed.
6. Finally, designers create the engineer's estimate. Designers manually enter all pay item related information again into the preliminary estimate of cost and bid schedule with its own format.

TxDOT follows similar principles for managing pay items but with slightly different procedures (3, 4). Unlike ITD's procedure, pay items are not identified by a capsule on

the plan sheet. Instead, all pay items for a given project are incorporated into the summary sheets. A standard sidebar summary on the plan sheet is not required. The designer has the flexibility to determine what kind of pay item summary should be included on the plan sheet, or if it should be included at all. Nevertheless, the entire process for managing pay items still primarily relies on manual operations.

NJDOT developed a program called EDQ (Estimate and Distribution of Quantities) for the purpose of improving the process of managing pay items. At NJDOT, circle capsules are used to identify pay items. A group of “To-be-Constructed” (TBC) boxes are drawn on the plan sheets to function similarly to the sidebar summaries used at ITD. The engineers use Excel spreadsheets to document and enter the quantities from the plans. These quantities are then entered into the EDQ program manually. The EDQ program can place the quantities, item numbers, and labels into both the project summary sheets and in the TBC boxes on the plans. However, the engineers still need to keep track of the pay items manually. Once a change is made to a plan, they need to modify the Excel file, provide the new quantities to the drafting technicians, who input the new quantities to the EDQ program, and make all the changes on the plan sheets. As a result, human related data entry and error management cannot be effectively controlled.

It is evident from the procedures described above that designers may need to enter the same pay item information several times at multiple locations. Designers have to manually keep track of all pay items associated with a design as plans are developed. Errors are inevitable, and it is hard to identify or locate them. As plans and drawings become more and more complex, identification of such errors is even more difficult.

The pay item related error is one of the primary causes of discrepancies between the contract plans, summaries and estimates. A significant amount of staff time is needed for engineers to create and double-check these documents. It is clear that an automated tool is needed to eliminate pay item errors in the design process and to increase the efficiency of the overall process. It is for the above reasons that ITD initiated this research project to develop API, the automated software tool that tracks and manages pay items in the highway design process.

3. Development of API

Design and Functionality Requirements

The development of API software was designed to meet ITD's specific requirements, which are described below for each of API's components.

- Plan sheets should be created inside MicroStation using ITD's standards and title blocks. Pay item information should be identified on the plan sheet by a capsule, which is an oval shape dimension element. The capsule with a leader or without a leader should be created by one command, which is adjustable until confirmed. This allows the designer to place the capsule in a place that is easily read, but not cluttered with too much information.
- A database should be established for storing the pay item information. After confirming the capsule placement, the basic information about this pay item should be copied into a database table strictly associated with the project while the capsule is attached to this row in the database table. The basic information should include the design file name, sheet number, pay item number, pay item name, pay item unit, start and end stations, quantities, and other relevant notes. The designer should be able to

edit the database information or delete the capsule with the row in the database table by graphically identifying the capsule. The pay item control file should be stored in a database table. The designer should be able to retrieve each pay item information from this table while working with API.

- The sidebar summaries should be created by a single command once the designer approves all the pay items on a plan sheet.
- The roadway summary and the bridge summary should be created automatically by a single command from the information stored in the database.
- The engineer's estimate report, which can be an ASCII format file, should be generated by one command. This ASCII file should be able to be used to create the preliminary estimate of cost and the bid schedule.

API's Workflow

Figure 4 depicts API's workflow and its major functions. In general, API stores the pay item information in a database, connects pay item information to a graphic element, automatically changes pay item summaries and reports, and creates the pay item report for the engineer's estimate. API allows a single entry of pay item information, and thus eliminates the errors associated with the existing multiple entry manual method. The MicroStation Development Language (5, 6) was used for coding API. An Oracle database was chosen for the initial UNIX version of API (7), but was later migrated to Microsoft Access for the Windows version (8).

Figure 5 shows the drop down menu of API and Figure 6 shows the dialog box for creating a pay item.

4. Field Testing of API

In 1998, ITD conducted a test to determine how effective API could be (1) to reduce the time required to prepare a pay item summary and (2) to reduce the number of errors associated with this task. Two experienced highway designers, one from ITD District 3 and the other from ITD District 4, conducted the tests with design projects of low to medium complexity. Each designer developed project plan sheets with and without API, noting the time required to complete each pay item summary. Comparisons of the number of pay item capsules, the number of entries in the sidebar summaries, and the number of entries in the roadway summaries were also conducted.

The results of the test are summarized in Table 1 and Table 2. Table 1 shows that, using the manual method, designers produced 8.7 and 1.3 pay items per hour, respectively.

However, using API, the designers were able to produce 17.5 and 3.8 pay items per hours, respectively. Overall, when using API, designers were able to produce 2.2 times more pay items per hour when compared with the manual method.

Table 2 shows that the error rate was greatly reduced when using API. Consider the data from both designers, the number of errors identified when comparing the pay item capsules and the sidebar summary data was 13 when the designers used the manual method and 1 when using API. The number of errors identified when comparing the sidebar summaries and the roadway summaries was 3 when the designers used the manual method and 1 when using API. In fact, the only errors that occurred while using API were caused by ITD's bid numbering system but not API itself.

These results are also summarized graphically in Figure 7 and Figure 8. On average, a designer can create 11 pay items per hour by using API compared to 5 pay items by the

traditional manual method, which indicates a more than 100 percent improvement in efficiency.

Subsequent beta testing was conducted in 1999 with at least one designer performing the testing in each of ITD's six districts. The time for the beta testing depended upon the size of the project used for the testing. Existing projects were used, with pay items being duplicated on copies of the project files. The beta testing successfully identified minor problems with functionality, needed enhancements, needed clarifications and/or changes to the user's manual. Most of the issues found or noted during the beta testing were minor in nature and were easily corrected.

API is now undergoing a major update as part of ITD's move to a new version of Microstation.

5. Summary and Conclusions

This study focused on the development of API, a CADD based software tool that operates within MicroStation. The software tool can be used by highway designers to record and summarize pay item information as part of the highway design process. API was specifically developed based on ITD's workflow, their hardware, software and operating system requirements.

Some of the major findings and conclusions from this study are summarized below:

- API maintains pay item information for a set of design plans in a central database. All pay items and their related attributes are selected and entered through a single and easy to use dialog box. All reports, including graphical summaries on the project plans, are derived directly from the data entered. This effectively

- eliminates errors and significantly reduces the efforts involved in the pay item data entry when performed manually.
- API greatly reduces the amount of time spent reviewing the sidebar summary and the roadway/bridge summaries. This automation removes one of the prime causes of discrepancies between contract plans, summaries, and estimates. It also reduces the large workload necessary to create and double check these documents, which are typically prepared separately.
 - An initial field test using API indicated significant cost savings due to the automated features of the software. The test of API showed a more than 100% improvement in efficiency. API also resulted in no pay item errors compared to an average of 16 errors when the process was completed manually.
 - Surveys of practice at both TxDOT and NJDOT on pay item management procedures revealed similar principles and procedures to ITD. However, each DOT's process and plan format are quite unique. There is a potential to extend API's applications to other DOTs, but would involve modification of the software structure to fit each DOT's specific needs.

Acknowledgements

The authors would like to thank the following individuals who have provided technical assistance and insights during the course of this research. Ms. Billi Miles of the Idaho Transportation Department worked very closely with NIATT staff and ITD project manager Wade Christiansen during this project and provided significant feedback to the development of API. Mr. Bruce Cosaboom at New Jersey Department of Transportation,

and Mr. Barry Heard at Texas Department of Transportation also provided important assistance.

References

- (1) Oglesby, C.H., and Hicks, R.G. Highway Engineering. John Willey & Sons, 1982.
- (2) ITD Design Manual, Idaho Transportation Department, 2003.
- (3) Design Division, Texas Department of Transportation. Plan, Specifications and Estimate Preparation Manual, 2000.
- (4) Design Division, Texas Department of Transportation, Project Development Process Manual, 2000.
- (5) MDL Programmer's Guide, Version 5, Bentley System, Inc. 1993.
- (6) MDL Functions, Version 5, Bentley System, Inc. 1993.
- (7) Yang, Z.X. and M. Kyte. API User's Manual, Version 1.0, University of Idaho, 1993.
- (8) Kyte, M. API User's Manual, Version 2.0, University of Idaho, 2000.

Table 1. Pay item efficiency test

	District 3 test			Number of pay items	District 4 test		Number of pay items	Overall	
	Number of pay items	Time required (hrs)			Manual method	API		Manual method	API
		Manual method	API		Manual method	API		Manual method	API
Roadway summary	92	4	1	-	-	-	-	-	-
Plan sheets	276	38	20	-	-	-	-	-	-
Total	368	42	21	15	12	4	383	54	25
Pay items per hour	-	8.7	17.5	-	1.3	3.8	-	7.1	15.3

Table 2. Pay item error test

District 3 test				District 4 test				Overall			
Pay item capsules vs Sidebar summary		Sidebar summary vs Roadway summary		Pay item capsules vs Sidebar summary		Sidebar summary vs Roadway summary		Pay item capsules vs Sidebar summary		Sidebar summary vs Roadway summary	
Manual method	API	Manual method	API	Manual method	API	Manual method	API	Manual method	API	Manual method	API
10	0	3	0	3	1	0	1	13	1	3	1

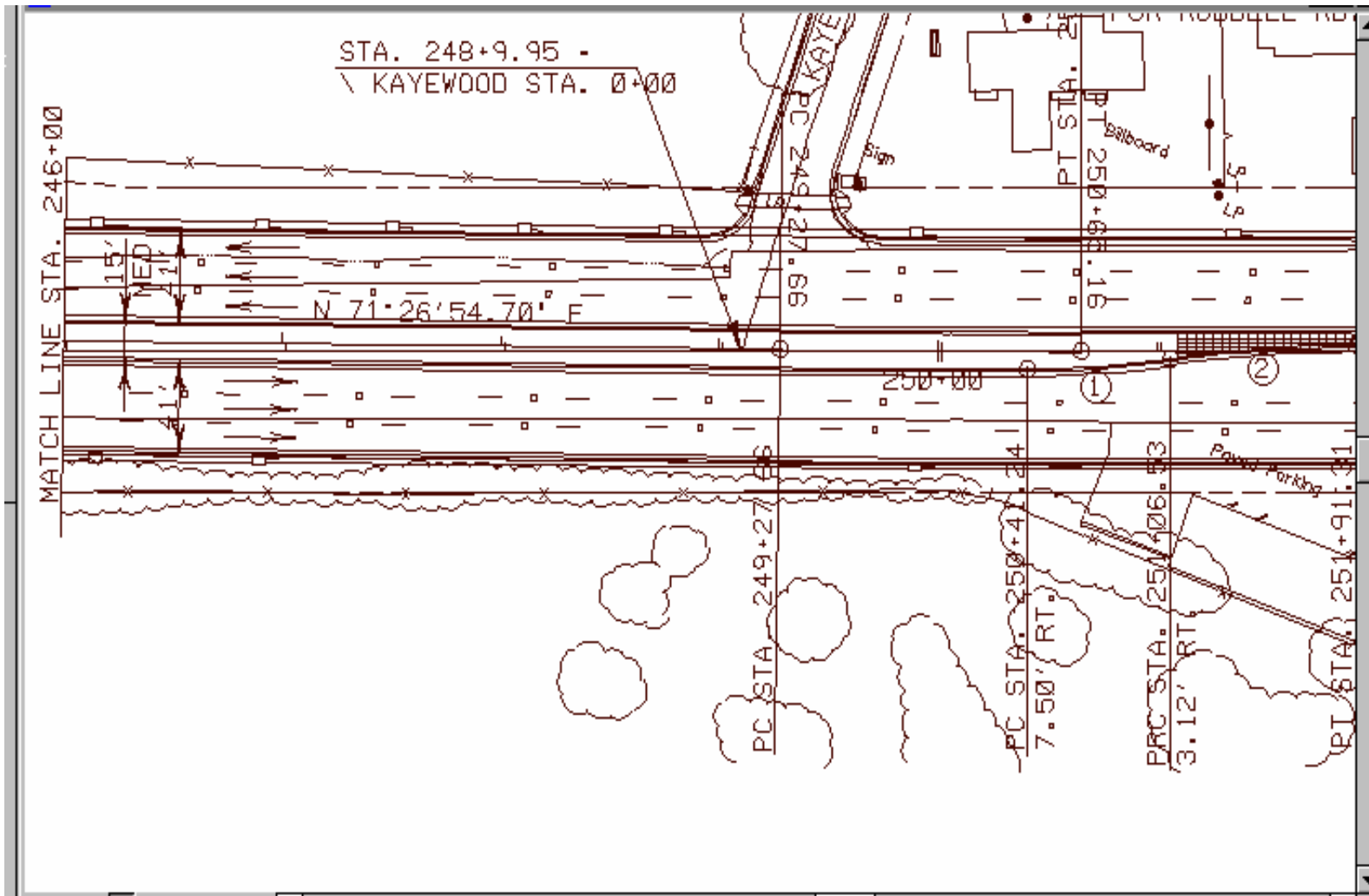


Figure 1 ITD Sample Plan Sheet without Pay Items

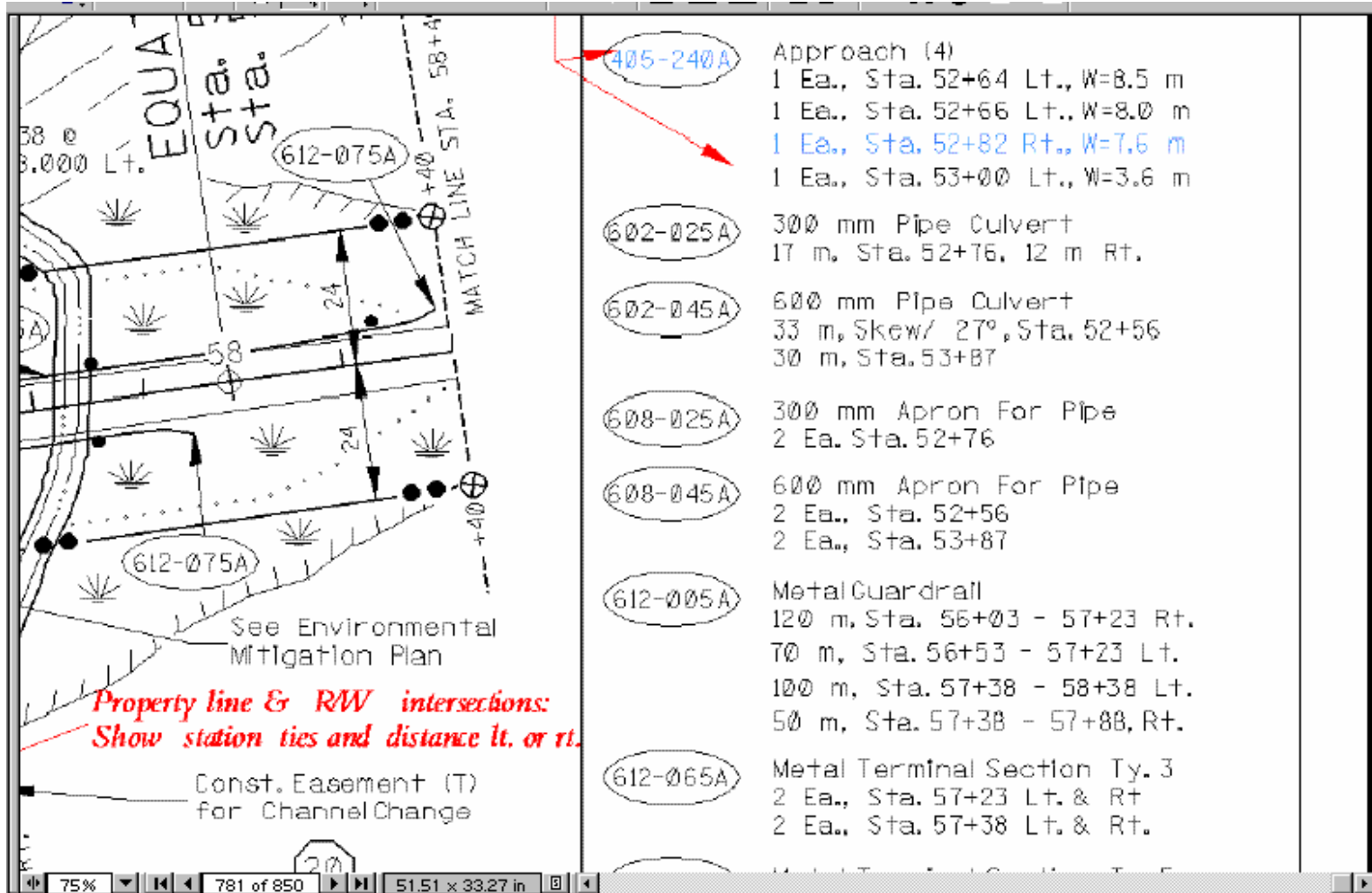


Figure 3 ITD Sample Plan Sheet with Pay Items and Sidebar Summary

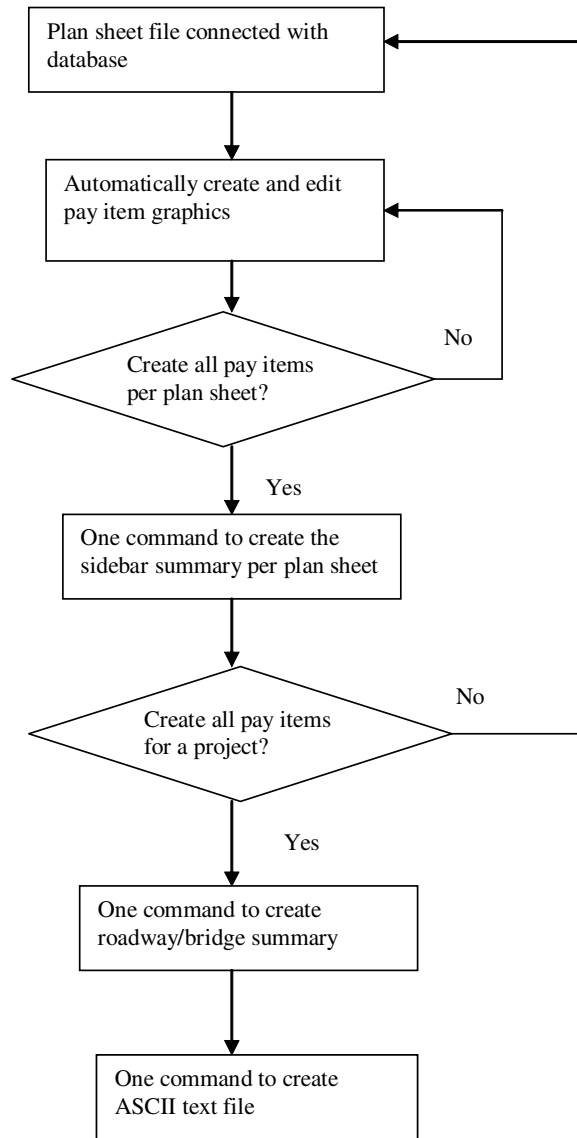


Figure 4 Work Flow for API



Figure 5 API's Drop Down Menu

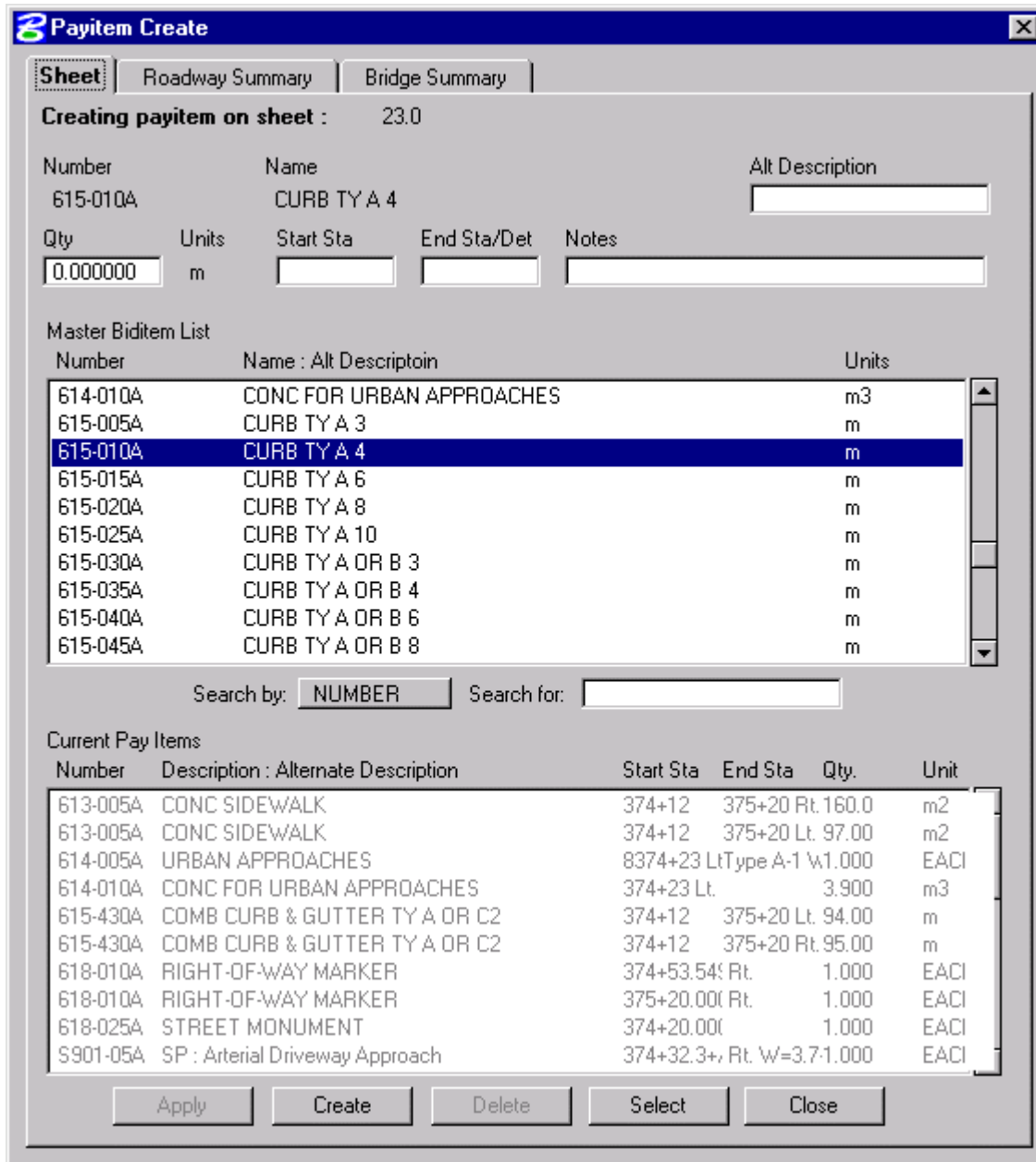


Figure 6 Create Pay Item Dialog Box

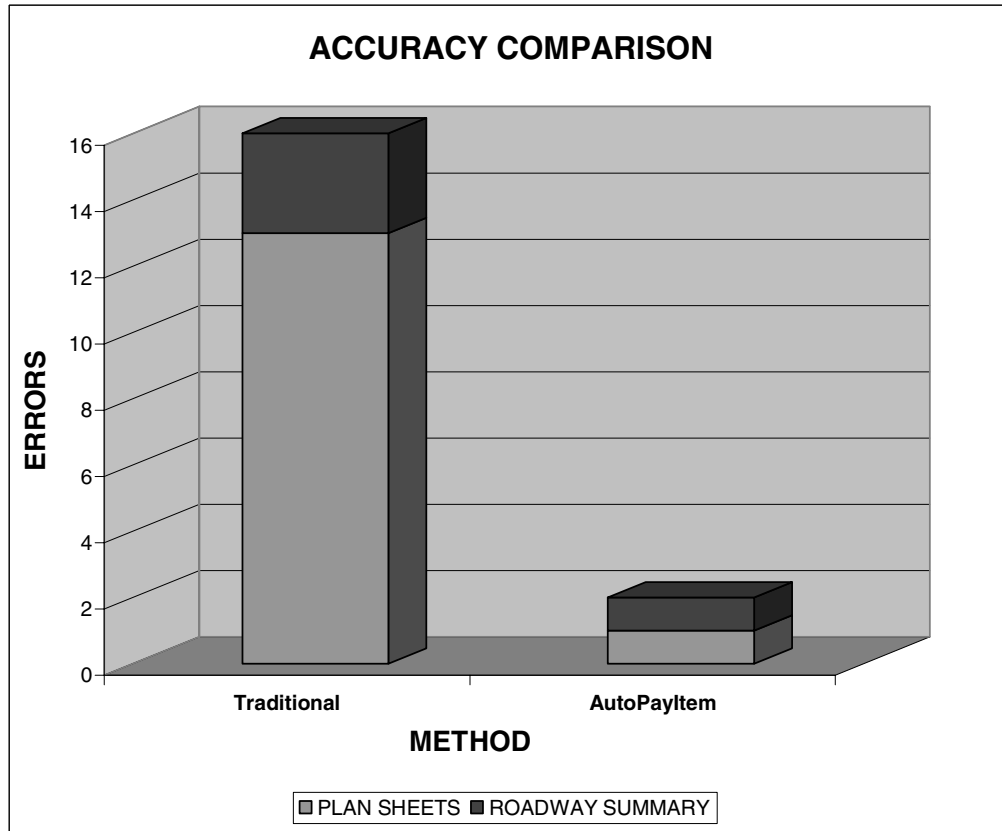


Figure 7 Comparison of Accuracy between Manual Method and API

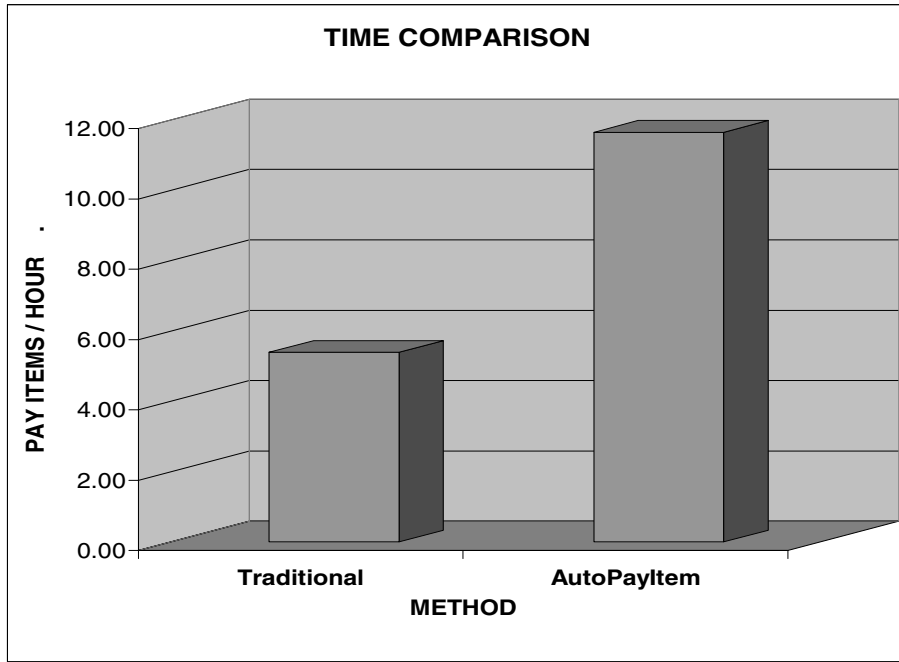


Figure 8 Comparison of Time Consumed between Manual Method and API