ASSUMPTION UNIVERSITY(ABAC), BANGKOK

มหาวิทยาลัยอัสสัมชัญ

A Model of Customized Application Integration based on SAP R/3 system

By

Supaporn Wongwithit

Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Science
in Technology Management
Assumption University

September, 2002

A Model of Customized Application Integration based on SAP R/3 system

by

Supaporn Wongwithit

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Technology Management

Assumption University

September, 2002

The Faculty of Science and Technology

Master Project Approval

Project Title A Model of Customized Application Integration based on SAP R/3 system

By Ms. Supaporn Wongwithit
Project Advisor Dr. Wisanu Tuntawiroon

Academic Year 1/2002

The Department of Technology Management, Faculty of Science and Technology of Assumption University has approved this final report of the three credits course. **MT6900** Master Project, submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology Management .

Approval Committee:

(Dr. Wisanu Tuntawiroon)
Advisor

(Dr. Soonthorn Pibulcharoensit)
Program Director

K. Phothikitti
(Dr. Kitti Phothikitti)
Committee Member

T. Sortrakul
(Asst.Prof.Dr. Thotsapon Sortrakul)
Committee Member

Faculty Approval:

(Dr. Soonthorn Pibulcharoensit)
Program Director

(Asst.Prof.Dr. Pratit Santiprabhob)
Dean

September/ 2002

ACKNOWLEDGEMENTS

This paper was written within the framework of customized application integration based on SAP R/3 for which it intensively required data support from other companies that have implemented the application integration and the data of the new technologies of application integration that exist in the market.

I am grateful to my supervisors Dr. Wisanu Tuntawiroon for being deeply committed to my work and for providing continuous guidance through the whole process of writing the report and other committee members: Asst. Prof. Dr. Thotsapon Sortrakul, Dr. Soonthorn Pibulcharoensit, Dr.Kitti Phothikitti.

I would like to thank all of my friends who brought me to study here and who supported me along the way I'm studying here. I would also like to thank all my colleagues who support me for studying master degree program and thank to my relatives for their encouragement and support as well. I would like to thank Dr. Wisanu Tuntawiroon again as the advisor of this project for frequent immediate advice and support during my studies studying here. Last but not least, sincere thank must be accorded to my beloved father Sutee Wongwithit and mother Supa Wongwithit for bearing with me through a long period of my life.

ABSTRACT

Enterprise computing has grown in just the last few years. Especially with the advent of the Web, not only is it possible for diverse organizations to automate and integrate their businesses and computer operations, it is imperative that they do so. Many corporations have become Web-enabled and are finding chat as they rely on myriad of applications; the ability to evolve and integrate existing application becomes significant. To be Web-enabled to support customers demand and third parties such as suppliers, dealers and so on.

Application Integration is part of the natural evolution of application delivery that includes improved software and the increasing acquisition of package software. It is the fact that companies need to integrate new applications with existing applications at a low cost and with minor change in. business processes. Many companies cannot afford to make such changes or discard existing systems.

With this limitations above, this project will discuss about a new model of Customized Application Integration to integrate information and system into one model at a low cost.

This model is simulated by SAP R/3 system with other systems that can be legacy system, or Web based system. The SAP R/3 system is the one of Enterprise Resource Planning applications mostly used in many companies around the world. To implement SAP will cost around 10 Million Dollars, so those companies who have SAP R/3 system don't want to discard the SAP R/3 system but want to utilize their business process that is processed by SAP R/3 system to integrate with new application systems such as Web based application.

This project will simulate the model that can integrate SAP R/3 with other systems with minor changes to the existing system and new applications at a low cost compared with Enterprise Application Integration in the market such as CrossWorlds, SeeBeyond or TIBCO.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

Nowadays, information is very important for everyone, especially for businesses because information can be the one source that leads business to gain a more competitive advantage than competitors. So many companies are trying implement new technology in their companies to be the channel to get information direct from customers or suppliers. As fast as they get information they can analyze and adapt their strategies to compete with competitors as the proactive company and to be the leader in providing products/service to customers.

Many companies become Web-enabled to support customers demand and third parties such as suppliers, dealers and so on. Web-enabled has very fast growth in the market. Many consulting firms believe that the market of e-commerce has explosive growth potential, as IDC projections indicate that business-to-business e-commerce revenue is expected to increase from $80 billion dollars in 1999 to over $1 trillion in 2003.

Most companies around the world use back-end applications like Enterprise Resource Planning (ERP) application. The popular one of ERP is SAP R/3 system. More than 1,000 companies around the world use SAP R/3 system. SAP R/3 system is the leading integrated enterprise software. It is comprised of four majors internal application categories, which are accounting, manufacturing, sales and human resources, containing more than 70 modules. SAP R/3 allows companies to automate or eliminate many costly and error-prone manual communication procedures, which

SAP R/3 provide functions that are easy to use in automatic works, and companies also can customize their own program to fit with the company's requirement as well. SAP R/3 can work for multinational corporations as well, since it can handle different currencies, difference languages, difference tax laws and regulations, and different requirements of several companies. SAP R/3 can handle and help to utilize excess capacity quickly and also reschedule when systems have many multifunction occurring.

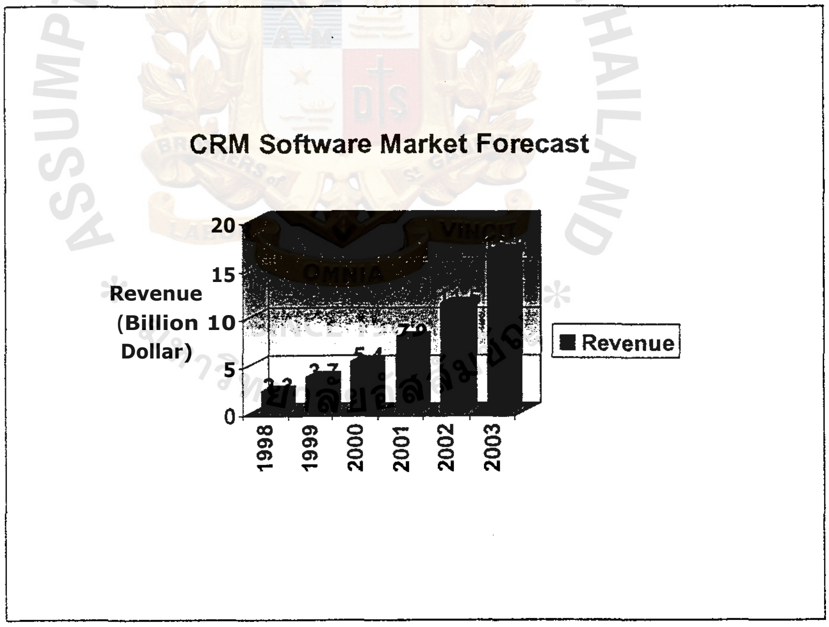
Implementing SAP R/3 is a difficult process and also maintenance, especially for large companies. Not only is it necessary to modify business procedures to conform to SAP's strict integration requirements, but SAP implementation is very complex and consequently very expensive like up to $200 million for a large company. For example, there are over 8,000 tables in the SAP R/3 database. These complicated tables direct the users through many menus and screens. SAP R/3 is based from a Germany company, so the database fields is the German language which is hard to understand. So to implement SAP R/3 system needs expertise from a consultant that will cost the company. Companies have the option to keep SAP R/3 system, with the high cost of implementation, an existing system and find the means to combine their functionality. In addition to retaining the existing systems, companies want to integrate them with new applications to enhance functionality to serve customers and suppliers needs.

Globally, more than 2,000 companies are planning major IT initiatives in the next three to five years. Some of the most popular include:

- Web-Based commerce
- Customer Relationship Management (CRM) application
- Supply Chain Management (SCM) application

- Business-to-business(B2B) commerce
- Customer self-service
- Expanding or integrating Enterprise Resource Planning (ERP) systems
- Mergers or acquisitions
- Compliance with major regulatory changes
- Business intelligence initiatives
- Mobile computing

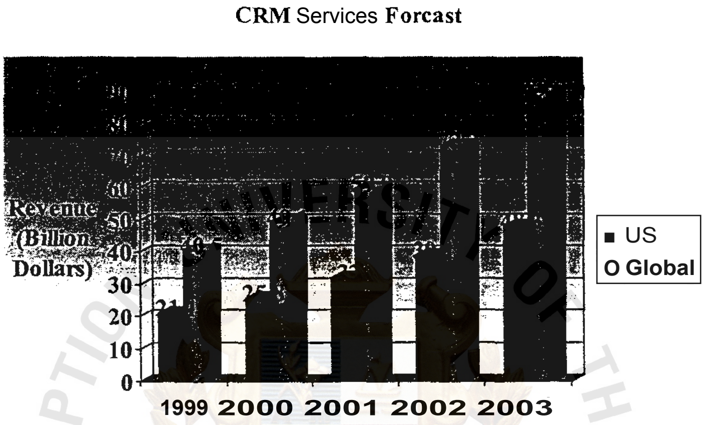
The two hottest applications in the extended enterprise market are Customer Relationship Management and Supply Chain Management. These two applications have high growth potential in the market regarding below figures:



Sources: ARM Research

Figure 1-1 Customer Relationship Management Software Market Forecast

As shown in Figure 1-1 Customer Relationship Management Software Market Forecast above, the CRM packaged software market is expected to continue expanding dramatically. The growth rate is expected to increase from $2.3 billion in 1998 to almost $17 billion by 2003.



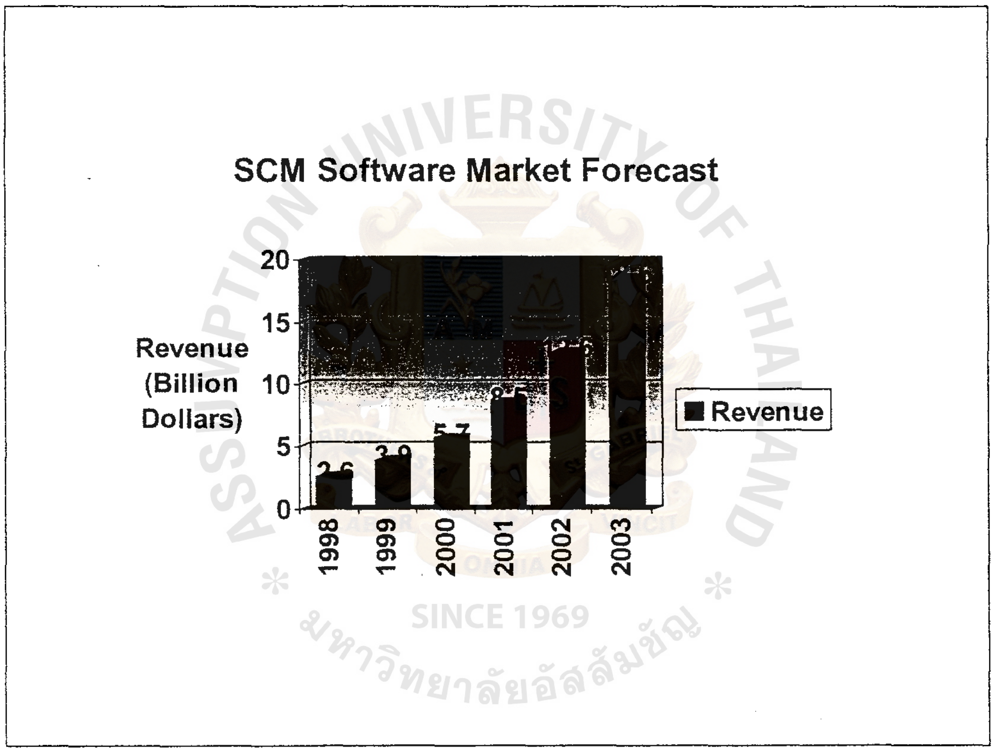
Sources: IDC and Cherry Tree & Co. Research

Figure 1-2 Customer Relationship Management Service Forecast

As shown in Figure 1-2 Customer Relationship Management Service Forecast above, CRM-related services represent a growing opportunity. From estimates through 2004, 80 percent of enterprise-level CRM initiatives will be outsourced to External Service Providers, and IDC forecasts that the global CRM services market, including consulting, system integration, outsourcing and training, will reach nearly

$90 billion by 2003. The high-end of CRM services market is currently dominated by the usual systems integration giants such as Andersen Consulting, Deloitte & Touche, PricewaterhouseCoopers, and the like.

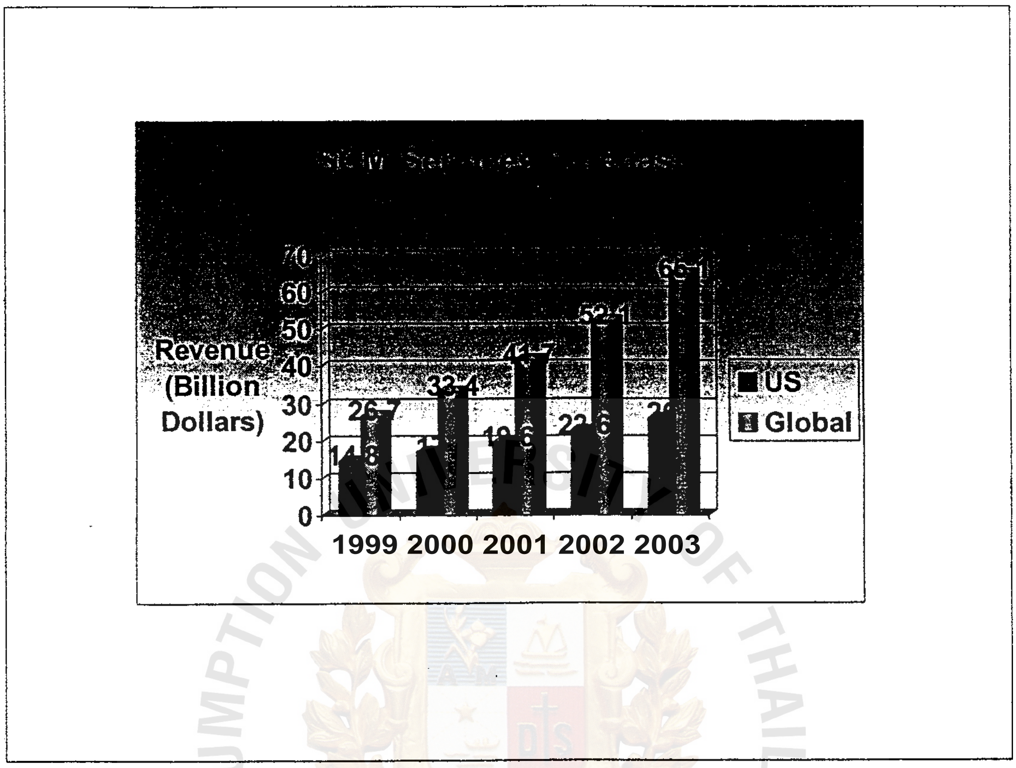
As is the case in the CRM market, demand for Supply Chain Management (SCM) applications is expected to continue its rapid growth. It is expected to expand at a compound growth rate of almost 50 percent and reach an estimated total value in excess of $18 billion by 2003 as figure below:



Sources: ARM and Cherry Tree & Co. Research

Figure 1-3 Supply Chain Management Software Market Forecast

The market for supply chain solutions, much like the CRM market is a case of too much demand chasing too little talent. In addition to the enormous challenges associated with integrating web-based back-end application with legacy mainframe and client/server systems. The figure below shows the exploding demand for SCM services provided by Cherry & Co Research.

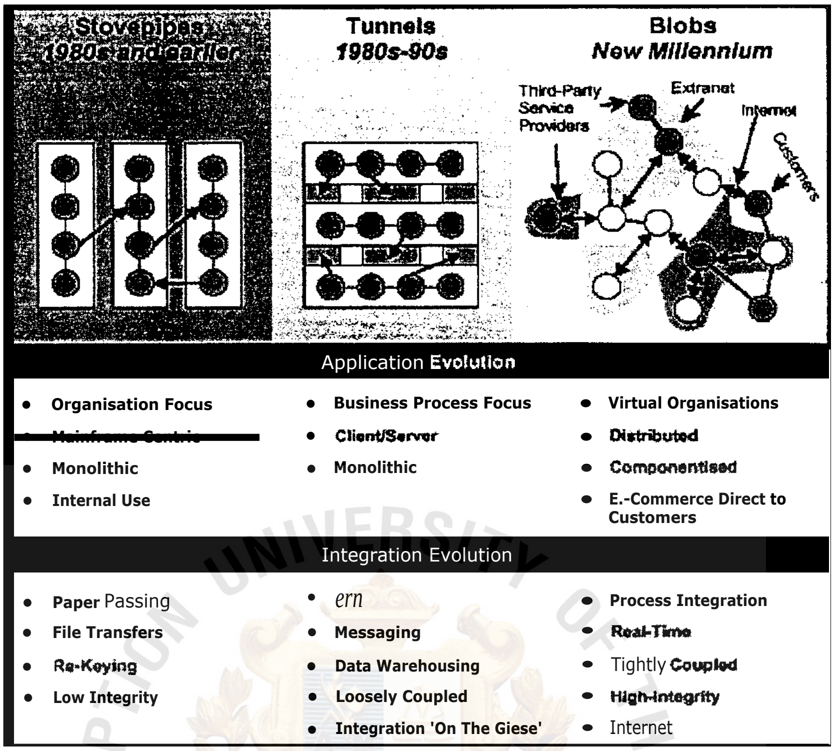


Sources: IDC and Cherry Tree & Co. Research

Figure 1-4 Supply Chain Management Service Forecast

So this would be a great opportunities for integration application providers to implement the way to integrate system in those companies. The integration market has many providers such as CrossWorlds, SeeBeyond, TIBCO, webMethods, Virtia and others, which is the part of Enterprise Application Integration (EAI).

Enterprise Application Integration entails integrating applications and enterprise data sources so that it can easily share business processes and data. Many of the requirements for integration can be summed as the figure below shows:



Application Evolution		
• Organisation Focus	• Business Process Focus	• Virtual Organisations
• ~~Mainframe Centric~~	• Client/Server	• Distributed
• Monolithic	• Monolithic	• Componentised
• Internal Use		• E.-Commerce Direct to Customers

Integration Evolution		
• Paper Passing	• ern	• Process Integration
• File Transfers	• Messaging	• Real-Time
• Re-Keying	• Data Warehousing	• Tightly Coupled
• Low Integrity	• Loosely Coupled	• High-Integrity
	• Integration 'On The Giese'	• Internet

Sources: Butler Group

Figure 1-5 Application and Integration Evolution

In the Stovepipes and Tunnels era's, whilst integration within applications might have been tight, integration between applications was much looser and, consequently, the real-time integrity of data was often weak. In the new millennium, applications will become much more like amorphous blobs that are much more difficult to manage, as many of their parts might be beyond control and in the hands of suppliers, paltners and customers, so integration through interfaces is the only way to bring an application together. In implementation, application integration has many complexities in an integrated system. So an appropriate architectures overview needs to be considered.

The general way of architectures can be separated into four key layers which are:

1. *Business Process:* This level needs to consider integration requirements and understanding of the sequence of events that triggers integration between components and applications and the role each plays in the overall business process.
2. *Business Objects:* This level needs the alignment of, or transformation between the business objectives contained in the components and applications.
3. *Business Interfaces:* This level needs the alignment of, or transformation between the interfaces and exchanges of information between the components and applications.
4. *Interface Technology:* This level needs the alignment of, or transformation between the technologies used to implement the business interfaces and achieve the integration.

EAT face a number of challenges with the changing requirements of customers or the system such as e-Business. These include the need to integrate legacy applications that are too costly, the requirement to aggregate information from disparate sources, the need to integrate disparate application systems for seamless information flow, time to market and so on. Many companies are concerned more with investments. So this issue is a big issue that EAI providers need to be concerned about.

Integration cost can be separated into three components: architecture, integration and operations.

1. *Architecture costs:* Architecture costs are capital costs related to the initial deployment such as the integration development, execution and operation environments. Architecture costs include the license cost negotiated with the vendor, the cost of new hardware required for integration, and the cost to implement architectural software and hardware. Roughly 80 percent of architecture costs are incurred for hardware or licenses as usage spreads.
2. *Integration development costs:* Integration development costs are separate from architectural costs and are often capitalized. They include development of interfaces and collaborations between systems. These costs are variable and driven by the number of interfaces developed.
3. *Operating costs:* Operation costs are expensed and include ongoing operations and maintenance of the EAI system for architecture and integration.

Consider a fortune 2,000 company that was planning to integrate its ERP system with a business portal, a project that required 190 interfaces. Architecture costs for an EAI solution based on CrossWorld's software amounted to $1,630,000, including $700,000 for software licenses, $90,000 for hardware, and $840,000 for architectural implementation. The cost of architecture for the custom solution is only $802,200 since software license fees are eliminated and implementation costs are lower.

The cost of developing reusable interface frameworks, such as request/reply for portal interfaces, is an example of an activity that's part of custom integration architecture costs. The primary EAI advantage comes in the area of interface development. Each custom interface takes 6.75 days to analyze and design, 18 days to perform detailed design, build and test, and 7.25 days for system test – for a total of 32 days. At a blended application development rate of $1,000 per day, this amounts to a cost of $32,000 per interface. Designing interfaces for reuse takes 30 percent longer, so analysis and design time is 8.78 days for the EM approach.

On the other hand, the EAI product used in this application, CrossWorlds, provides a 25 percent productivity gain in development relative to custom approach, so detailed design, build and test time is reduced to 13.50 days. System testing time remains the same at 7.25 days. Total time to build each interface adds up to 29.525 days for a total cost of $29,525. An even more important difference with EM is that 42 of the 191 interfaces involved in the project can be reused. Of course, reuse isn't free, but on this project, interfaces that were reused were 80 percent less expensive than the ones that had to be developed, with the remaining cost primarily consisting of systems testing time. The custom architecture development cost was $420,000 while the EAI architecture cost was $445,916 due primarily to license and hardware necessary to integrate the new ERP instance. The total cost for the project of EM is $6,525,245. This can summarize as in the table below :

	EAI
First Project	
Architecture Costs	
Software licenses	$700,000
Hardware	$90,000
Architectureal implementation	$840,000
Total	$1,630,000
Interface DevelopmentTime	
Analysis and design time	8.78 days
Detailed design, build and test time	13.50 days
System test time	7.25 days
Total Time	29.53 days
Application development rate	$1,000/day
Cost per interface	$29,525
Nominal interface costs	$5,639,275.
Saving from reuse of interface	$744,030
Total	$4,895,245
Total Costs- First Project	$6,525,245

Source : eAI Journal

Table Integration cost of EAT base on CrossWorld application

The flexibility of EAI allows for changes to the business and technical landscape to occur with minimal rework and impact on production system. Increased manageability and maintainability provide extended technical control of the environment for proactive systems management. The EAT approach also provides improved access to more timely, accurate data across a distributed environment while minimizing redundancy.

1.2 Problem Statement

This era is the era of information and it is very important in doing business. Many companies try to gain more competitive advantage than competitors by implementing new applications to get information from customers or suppliers. So they try to use many applications that can reach to customers or suppliers directly to integrate with their Legacy system such as e-Business, Customer Relationship Management (CRM), Supply Chain Management (SCM) and so on.

World wide, there are more than 1,000 companies that have SAP R/3 systems to be the back-end system that cost a lot in implementing the SAP R/3 system. So those companies might not want to discard the SAP R/3 system but they need to implement new applications integrated with their SAP R/3 system with a minor change on functionality of their business. They need the application integration to enhance their functionality to serve customers and suppliers need.

The integration market also has many providers but it is very expensive. So many companies cannot afford for the new investments at a high cost but they also need to compete with their competitors in the new era.

1.3 Goal and Objectives

- To identify the basic need of Application Integration based on SAP R/3 and other systems such as e-Business and Legacy system.
- To provide the basic model of Customized Application Integration between SAP R/3 and other systems such as e-Business and Legacy system at a low cost.
- To provide the prototype program of Customized Application Integration between SAP R/3 and other systems such as e-Business and Legacy system.

1.4 Scopes and Limitation

This project discovers and learns about the basic need for application integration and provides the basic model of Customized Application Integration between SAP R/3 system and other systems. This project also provides a flowchart of the integration and also provides an example prototype program in integration, which includes inbound and outbound integration of SAP R/3 system that can re-use some source code for basic inbound and outbound architecture in integration between SAP R/3 and other systems. This project not only support companies that already have SAP R/3 as an existing system but it also supports new implementation of the SAP R13 system in integration with Legacy system. This project used basic model of integration with the transfer file mechanisms and this project also provides an interface for both of workstation and UNIX path in transferring data between systems.

The limitation of this project is real-time information because this project used transfer file mechanisms that is the cause for non real-time information but it is nearly real-time which support by SAP R/3's function but if it needs to use SAP R/3's function, it might cause another issue that is performance issues for SAP R/3 because program need to load the system all the time to trigger incoming files from other systems. Another limitation of this project is the number of connections between systems because this project need to have one to one connection for each system that include 2 programs which are inbound and outbound program. So if they have more than one system, they need to have another connection for the new connection.

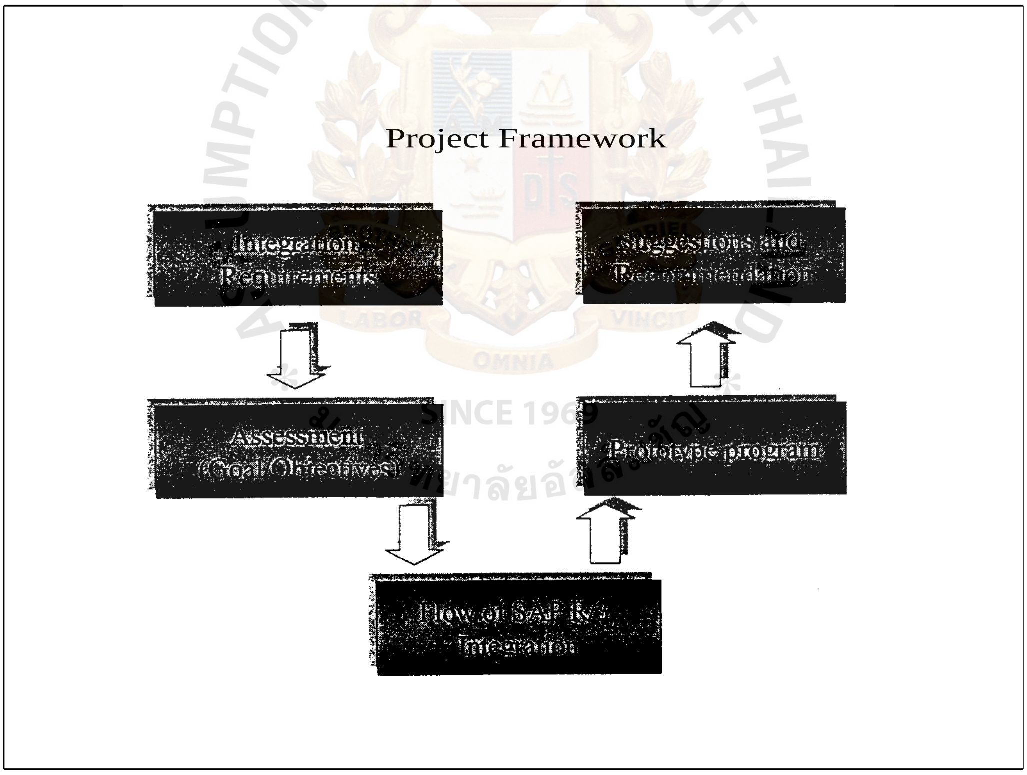

Project Framework
Integration Requirements
Assessment (Goal Objectives)
Flow of SAP R/3 Integration
Prototype program
Suggestions and Recommendation

Figure 1-6 Project Framework

1.5 Development Plan

Project Planning	N v, 20 N	Dec,2002	Jan,2002	Feb,2002	J Z00Z	Feb,2002	r,20OZ	A r,Z0Z N	May,2002	Z 02	2002	A Z0 N	Sep,2002
Pre-Feasibility (RPM)	■	■											
Research Study	■	■											
Project Feasibility Study	■	■			■								
Identification of Data Collection		■			■	■							
Design prototype of Integration Application						■							
Coding prototype program for Inbound and Outbound programs							■	■	■				
Review Project										■	■		
Register for project											■		
Writing Project Proposal											■	■	■
Project Evaluation													■
Conclusion													■

Table 1-2 Project Planning

2. LITERATURE REVIEW

2.1 Butler Group, "Application Integration", May 1999.

Butler Group focuses on application Integration, which Butler Group defines as the requirement to integrate into new business processes the functional behavior, or business rules of disparate systems, or components of them, as well as, but not just, the data that underlies them. The focuses in the past have been more on integration of in-house developed applications and components, which is easier when all of their source code is available and controlled within the project or same Information System (IS) department, and can be changed to enable integration.

Butler Group observes that Application Integration is effectively becoming a discipline in its own right, due to the following trends:

- Greater need for real-time integrity and process integration, not just data exchange and replication.
- New integrated business processes are not only crossing organizational boundaries within a company, but flow between many companies too.
- Constant introduction of new business processes, requiring re-integration of the same core business logic into new applications.
- The increasing need to integrate new application workflows with externally developed black-box software, of which implementations are hidden from the developer that can only use their existing interface.
- The need to integrate one package with another, where both are black boxes and the only development task is integration.

- It has become more complex technically. There are more technologies involved, and these are increasingly more complex to use.
- Reduced business change cycle times. Developers cannot respond quickly enough to integration needs by developing all the integration software in-house.
- The productisation of integrationware. The timely arrival of off-the-shelf solutions to common integration scenarios and the possibility to automate integration tasks.

Butler Group sees that as there is no time to build new applications from scratch, and no one package providing an ideal or complete match to requirement, then the only timely response seems to be to integrate whatever companies already have, or can quickly acquire or build.

Butler Group believes that Application Integration is a permanent state that requires architectural foundations, which enable continuous, efficient and rapid reaction to seemingly random events. Butler Group also provides the solution for a critical subject and provides frameworks for managers to communicate the critical issues that must be addressed to establish a reactive environment.

2.2 Cherry Tree & Co., "Extended Enterprise Applications", Jan 2000.

The extended enterprise is a business whose information system operates within distributed application architecture. This architecture is arguably the most critical component of the new e-business environment that IDC projections indicate that business-to-business e-commerce revenue is expected to increase from $80 billion dollars in 1999 to over $1 trillion in 2003. Given this explosive growth potential, Cherry Tree & Co. believes that the market for the extended enterprise applications that enable this e-business environment will expand dramatically.

Cherry Tree & Co. study about concept of the extended enterprise and explain its relevance to IT services firms. Cherry Tree & Co. emphasize into two of the hottest segments in the extended enterprise space which are Customer Relationship Management (CRM) and Supply Chain Management (SCM) to demonstrate how the functionality of corporate IT system is being extended beyond the enterprise.

Cherry Tree & Co. review the evolution of the technology, but the main thrust of the report will center on the enormous opportunities that have been created for External Service Providers (ESPs) by the increased demand for extended enterprise applications. Cherry Tree & Co. believe that with proper planning and execution, privately held IT service companies can be well-positioned to take advantage of this new wave of opportunity.

At the core of the extended enterprise site Cherry Tree & Co. emphasize on the core ERP backbone or other core accounting, manufacturing and HR applications. These applications reside within the enterprise and can be described as being primarily inward facing applications that track the internal flow of information. An enterprise starts to become extended when its information systems face outward by enabling connectivity with customers, suppliers and distributors. A company completes its evolutions and becomes a truly extended enterprise when this connectivity with its business partners becomes fully integrated into its ERP backbone.

Cherry Tree & Co. review how Enterprise Application Integration tools are being utilized in extended enterprise environments to enable connectivity of multiple applications both within and between companies.

2.3 Aberdeen Group Inc., "e-Business Infrastructure Integration: Practical Approaches", Nov 2001.

Aberdeen Group Inc. study about e-Business Infrastructure Integration because many companies perceived and realized economic benefits of e-Business in reduced cost of operations, improved levels of service and expanded market reach which have fueled a dramatic growth of innovative e-Business software application supporting new paradigms for customer-driven selling and self-service, electronic payments and vendor-managed inventory. Customers are also changing their buying habits and increasing their expectations for superior service. Intensifying competition between Web sites and escalating customer expectations continue to drive companies to differentiate themselves by exploiting the Internet to offer superior service and more efficient business-critical procedures.

Aberdeen Group Inc. study how to achieving these improvements via the Internet requires companies to integrate data and content between legacy, packaged and customer applications. Aberdeen Group Inc. faced the challenge of accomplishing integration at a rapid pace to keep up with the growth rate of new e-Business applications. However, the recent results of integration projects have produced higher costs, longer projects and higher levels of complexity than customers anticipated.

Aberdeen Group Inc. explores the technologies, methodologies and services that can reduce the overall risk, cost, and effort of integrating disparate applications and information sources. It also defines the requirements that must be met to provide successful e-Business integration. It then briefly reviews the technologies that should be considered. Moreover, Aberdeen Group Inc. discusses a practical approach and best practices for developing an integration project and reports experiences and feedback from customers who have completed an e-Business integration project.

2.4 SeeBeyond., "SeeBeyond Application", Oct 1999.

SeeBeyond provides leading eBusiness Application Integration (eAI) solutions for multiple vertical industries, including energy, financial services, government, healthcare, manufacturing, retail and telecommunications. These solutions span all areas of the business and encompass many different areas of technical functionality, but can be thought of as belonging to two primary domains - Supply Chain Integration and Customer Relationship Management (CRM) Integration. SeeBeyond models, manages, and integrates business processes within these domains and has the only unified network centric solution able to meet the requirements of the world's largest corporations.

Supply chain integration provides the infrastructure required to truly enable the collaborative supply chain by providing instantaneous global access to information. Whether the challenge be improving sourcing and procurement, achieving collaborative design and manufacturing objectives, implementation of a new supply chain management solution, or improving order management and fulfillment operations, SeeBeyond has the solutions and experience required to be successful. By enabling the visualization and modeling of business processes, all the way through to their implementation within a fault-tolerant, fully distributed architecture, only SeeBeyond provides a comprehensive, unified solution developed from the ground up by a single vendor.

Customer Relationship Management (CRM) Integration is the solution domain focused on improving revenue by identifying, acquiring and retaining the most valuable customers through a combination of customer insight, sales and marketing programs and customer support activities that exceed customer expectations.

Intuitively, SeeBeyond know that these activities require access to all available customer information when and where necessary. SeeBeyond also know this task is made extraordinarily difficult because this information is stored in many disparate systems, across geographies, and is frequently inaccessible and certainly not in a timely manner. Solving these problems of access to information, and more importantly real time access to information in the context of the business process, is the solution SeeBeyond provides solutions to marquee customers across multiple vertical industries in support of their Customer Relationship Management objectives.

2.5 **ACTel.,** "Case Study: **ACTel",** Sep 2000.

Company background:

ACTel is a leading telecommunications company headquartered in Switzerland. With 22,000 employees, the innovative, customer-focused group offers a full range of voice and data communication services on fixed-line and mobile networks. The company currently provides some 3.9 million analog access lines, as well as around 920,000 ISDN access lines, and serves over 1.6 million mobile phone subscribers.

Business Challenge:

To master the challenges of deregulation in a rapidly evolving market, ACTel needed to establish a strong and effective software infrastructure. In 1991, the company started looking for new software that would introduce greater cost transparency, streamline business processes to improve cost-effectiveness, and ultimately enable it to price its products and services more competitively. ACTel used SAP R/3 to be a core element of SAP's industry solutions. The following six core modules were implemented: Project Mangement (PS), Finance & Accounting (FI), Controlling (CO), Executive Information System (EIS), Material Management (MM), and Sales & Distribution (SD). Also, the company opted to retain its Global Account Management System (GAMS), a custom Oracle application, which maintains customer information and its Siebel system as its customer care application.

A team of consultants worked with ACTel to implement SAP and to build the interfaces between GAMS and SAP, GAMS and Siebel, and Siebel and SAP utilizing the CrossWorlds product suite. The integration would ensure the when new customers were added, or changes to customer information were applied in GAMS, the changes were synchronized with the SAP R/3 and Siebel systems. Also, the interfaces would allow a sales representative to create a Sales Order and receive order confirmation number.

Assumptions:

- Implementing R/3 for finance and material reporting.
- Retaining the Global Accounts Management System (GAMS).
- Implementing 12 for supply chain management.
- Implementing Ariba for managing maintenance, repair, and operating supply and procurement.
- Retained Siebel as the customer relation management application

Application Architecture:

<u>Data Flow /Integration Point Diagram</u>

This schematic diagram is a high-level summary of the major ERP system showing one single connector line between each ERP and other systems with summarized interface flows.

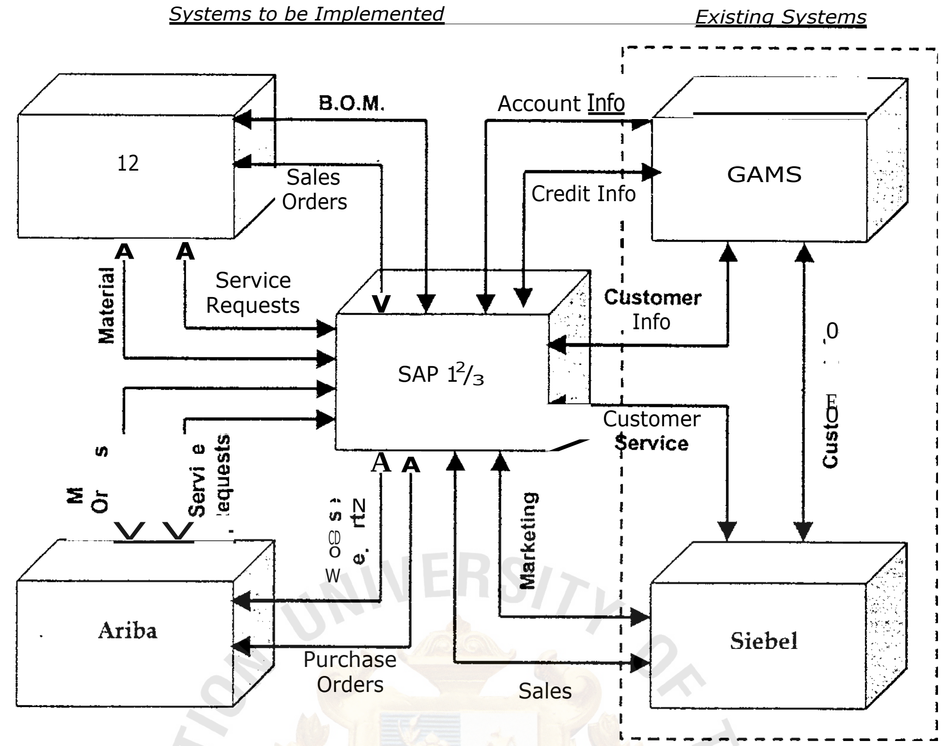


Figure 2-1 Integration Point Model

Building customer loyalty is vital to ACTel long-term profitability, and the integration of Siebel with the implementation of SAP will provide a comprehensive sales order creation and confirmation capability needed to achieve their goals. Another key capability is the synchronization of customer information within the Siebel, GAMS, and SAP applications. Any changes or additions made to the master customer database in the GAMS system will instantaneously be synchronized with the customer information in SAP and Siebel.

ACTel also opted to implement 12 system to optimize their supply and demand of inventory in order to minimize their product costs. By implementing Ariba, ACTel will also be able to streamline the cost of procuring items needed to do business. At the backbone of this comprehensive solution will be the SAP R/3 system, which will aid in managing all the information.

Implementing and integrating these systems into ACTel’s line of business would introduce greater cost transparency, streamline business processes to improve cost-effectiveness, and ultimately enable it to price its products and services more competitively.

Integration Point List:

The integration point defines the interface needed for each of the six core business processes. This table documents the different integration points, consisting of the business process, the content or data requirements, the source application, and the destination application.

Process	Data	Systems
Sales Order Creation and Confirmation	Sales Order, Confirmation Number	Siebel and SAP
Material Planning	Materials, Sales orders, BOMS	i2 and SAP
Customer Synchronization	Customer Info, Account Info	GAMS, SAP, Siebel
Expense Management	Expense reports	SAP and Ariba
Service Management	Service Orders	12, SAP, Ariba
Procurement	Purchase orders, MRO orders	Ariba, SAP

Table 2-1 Integration Point List

Business Capabilities:

Customer Relationship Management - Siebel will continue to be used as the primary application to support all customer care / relationship management activities. Sales orders will be entered directly into the Siebel application and interfaced to SAP for order processing. Given that this is a customer-facing activity and that availability & ship dates need to be determined immediately; this interface will need to be near "real-time".

Customer Information Management – the GAMS system (custom-developed Oracle applicaton) will continue to be used for the management of the customer master record. This information will be interfaced to all other relevant applications (SAP, Siebel) for master data management. Updates to the GAMS information will be shared with the other applications via batch processing multiple times per day.

Inventory of Integration Components:

The Application Architecture Diagram identified several integration points that need to be created. These integration points contain different data layouts, which will be the driving components for interface architecture. The data layouts and the data sources are documented in the table below for the development team to use in the creation of business objects and the corresponding mappings and transformations.

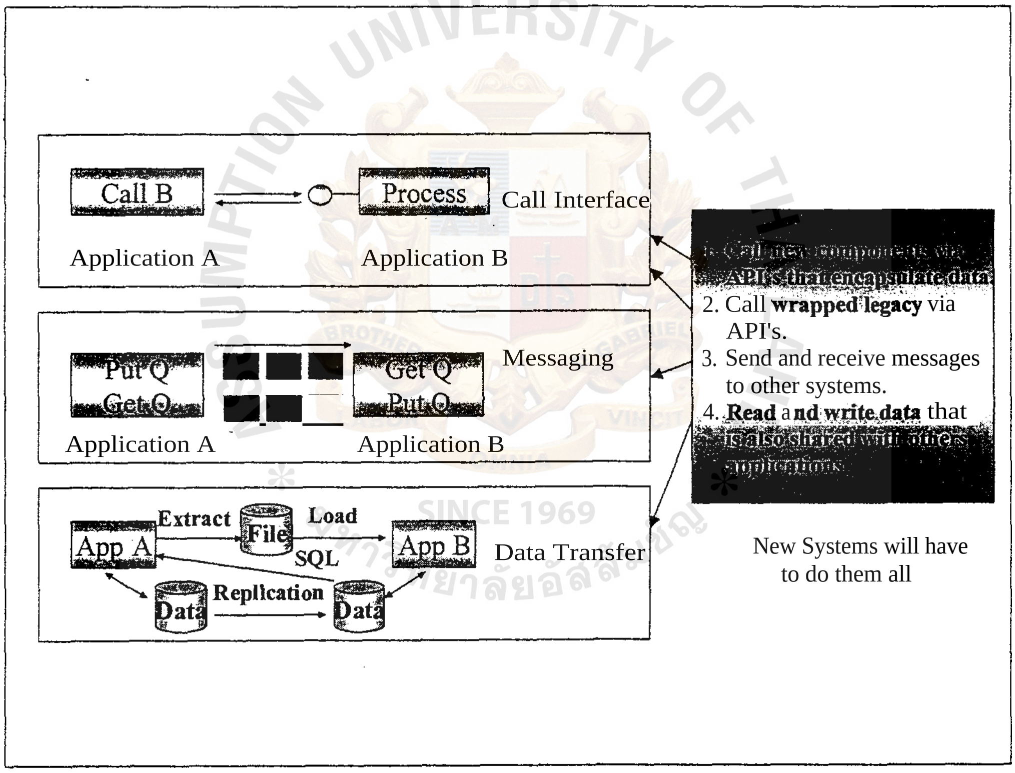
Integration Components	Data Source
Sales	Siebel and SAP
Marketing	Siebel
Customer Information	GAMS
Customer Service	Siebel
MRO order	Ariba
Material	12
Service Request	12 and Ariba
Expense Reports	SAP

Table 2-2 Inventory of Integration Components

3. METHODOLOGY

3.1 File Transfer

The way to integrate data from difference applications can be separated into the three categories as the figure below shows which are Call Interface, Messaging and Data Access/File transfer.



1. Call new components via APIs that encapsulate data.
2. Call wrapped legacy via API's.
3. Send and receive messages to other systems.
4. Read and write data that is also shared with others applications

New Systems will have to do them all

Sources: Butler Group

Figure 3-1 Integration Mechanisms

Call Interface:

Applications provide a callable interface, usually referred to as an Application Programmable Interface (API). An example of this call interface is packaged application interface such as SAP's BAPI (Business API).

Advantages:

- Need to ensure the real-time integrity of transactions and data.
- Encapsulating the implementation behind a common interface.
- Need to invoke business logic, not just retrieve data.
- Building new components.
- Providing wrappers around existing system, which can often be achieved without changing the source of the system.

Disadvantages:

- Might be complex to program and new technologies.
- There are many different technologies, though reducing this factor is one of the major attractions of integration.
- Synchronous behavior requires the application and the connection between them to be up and running.
- Expensive.
- Performance concerns because it needs to load system all the time.

Messaging:

Applications are integrated by send and receive messages, usually via some queuing mechanism. For example Mail systems and groupware products such as Microsoft Exchange or Lotus Notes.

Advantages:

- Enables asynchronous, loose coupling of distributed application.
- Can be used for a publish and subscribe approach, where the sending application requires no knowledge of what applications subscribe to its messages.
- Lower cost of implement than Call Interface.

Disadvantages:

- Requires applications to use the messaging interface, and know when/how to read and write the queues, which requires the code of legacy application to be changed.
- Can require extra effort to add synchronous, real-time behavior on top of messaging system.

Data Access/File Transfer:

Applications are integrated direct access to their database, or via file transfers. For example file transfer include batch loads, direct read and write database using database calls and database gateways such as Information Builder SQL.

Advantages:

- Useful when there are large volumes of data to move.

- Supports off-line analysis and reporting on large volumes of data.
- Can be straightforward and easy to implement.
- Does not require the existing application being integrated to be changed.
- Cheapest cost when compared with the Call Interface and Messaging.

Disadvantages:

- Low integrity, as replicated data is out-of-date.
- Low integrity, if business rules and validation of existing application are bypassed.
- Does not encapsulate physical implementation and new applications are affected by change to the ones integrated.
- Data may require interpretation to be turned into information.

This project will simulate SAP R/3 system as a key driven to integrate with other systems that can be e-business or Legacy system by using transfer file methodology during integration of system that is explained in the figure below:

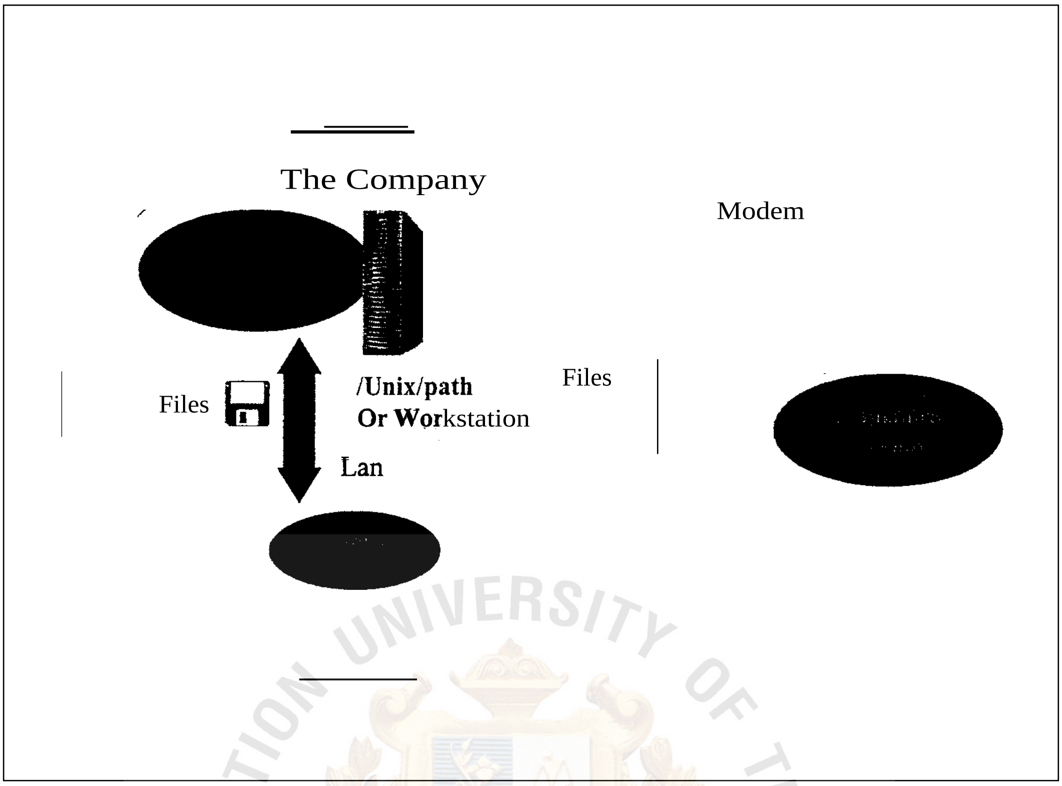


Figure 3-2 File Transfer Methodology of Customized Application Integration

This project uses file transfer Methodology to integrate with other systems via workstation path or UNIX path because it easy to implement and the main reason is file transfer methodology is the cheapest way in integration and this is emphasized in cost of implementation. Not only for the cost that is cheapest but file transfer is useful for the transfer of large amount of data between systems. And it supports for the batch job analysis or off-line analysis as well. File transfer needs a bit change for the existing system with minor change in functions and existing program code. For another new application that wants to integrate with SAP R/3 can use only file transfer to send data to SAP R/3, that is the easy way to integrate.

4. PROJECT IMPLEMENTATION

4.1 Overview

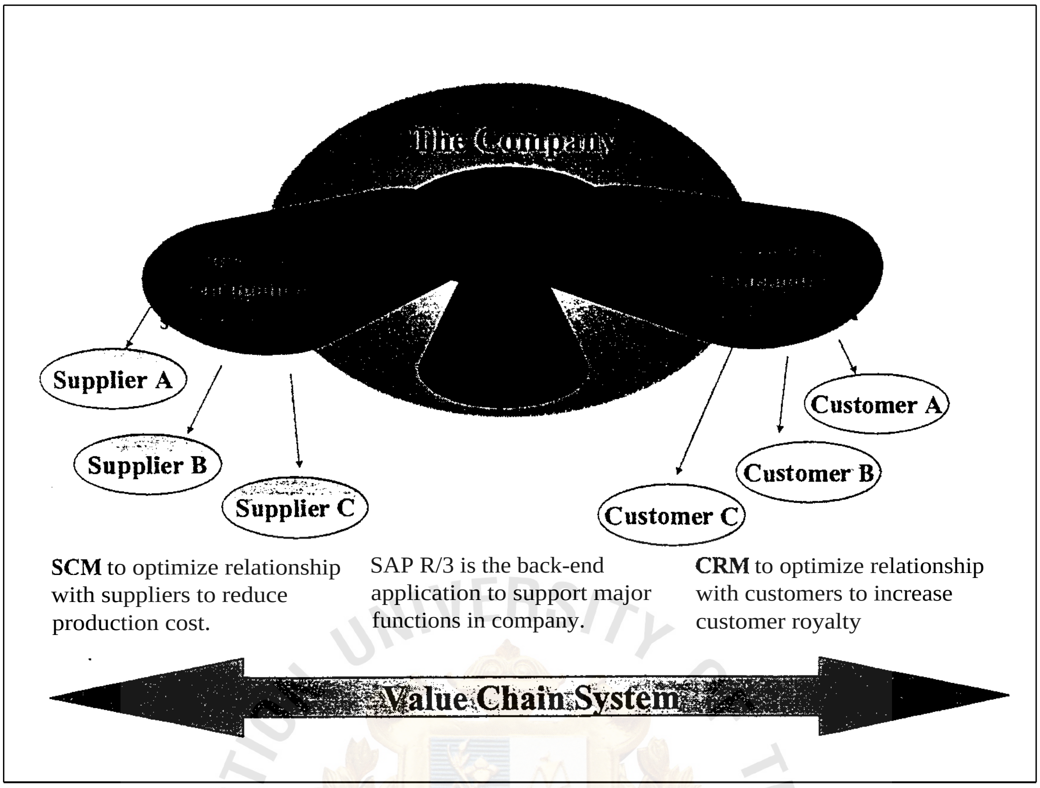
Many companies implemented the SAP R/3 system in integrating internal functions needed in their enterprise, where major application categories are accounting, manufacturing, sales and human resources. That might not cover for external demands and trend of Value Chain. So companies need to serve external demands such as customers demand and suppliers demand. They need to implement application that can enhance their capacity to compete with competitors or to expand their business to be a World Wide Company. So basic need for most companies is to integrate their SAP R/3 system with new applications such as Customer Relationship Management (CRM), Supply Chain Management (SCM) and so on. Not to integrate only new applications but the company need to become web-enabled to expand their market worldwide.

For the company, which is preparing to implement SAP R/3 to serve their basic functions in their company, they also need to integrate SAP R/3 with their Legacy system, as SAP R/3 can't support some functions of Legacy system.

But for the issue of concern for those companies who want to implement new applications to integrate with SAP R/3 or integrate their Legacy system with SAP R/3, is the cost of investment. Even though, in the market there are many Application Integration providers but it quite expensive so many companies can't afford for the investment. They try to find a suitable solution at a low cost in the market and also those companies don't want a big change in their existing functions.

4.2 Basic Model

The basic model of Customized Application Integration between SAP R/3 and other systems such as e-Business and Legacy system, mean that companies trying implement the new application to cover all Value Chain concept. Most of the companies who prepare to implement SAP R/3 or already have SAP R/3 for supporting major functions of accounting, manufacturing, sales and human resources. Many companies need to be like Value Chain that they can serve from suppliers to customers. So the main functions that are concerned with SAP R/3 are accounting, manufacturing and sales. These three functions will effect for the integration because it needs to integrate data with these three functions to serve customers and suppliers demand that can be explained as figure below:



SCM to optimize relationship with suppliers to reduce production cost.

SAP R/3 is the back-end application to support major functions in company.

CRM to optimize relationship with customers to increase customer royalty

Figure 4-1 Basic Model of Customized Application Integration between SAP R/3 and other systems

This model can support for the need that company want to enhance their products/services with suppliers or customers and this model also support for the Legacy system that has SAP R/3 as the key driven in the company.

4.3 Flow Chart

This project simulated main functions of SAP R/3 that are accounting, manufacturing and sales. These three functions were integrated with e-business system and Legacy system for the e-business system can have more than one application such as SCM or CRM applications and Legacy system will be integrated with SAP R/3 to serve the customization demand for the company. This model simulated for the company, which is the manufacturing company to produce that product to customers. This company will get the sales order from sales department, after that will planning for the material of each sale order and send request to SAP R/3 system: SAP R/3 system will maintain for the material such as good issues and good receipt of the material and generate document for the goods movement. SAP R/3 system will process for the accounting by maintaining for the financial of purchasing of each sale order and crated for the confirmation documents. After the SAP R/3 system generates documents in the system, these documents will be transferred to related system such as Legacy system and e-business system.

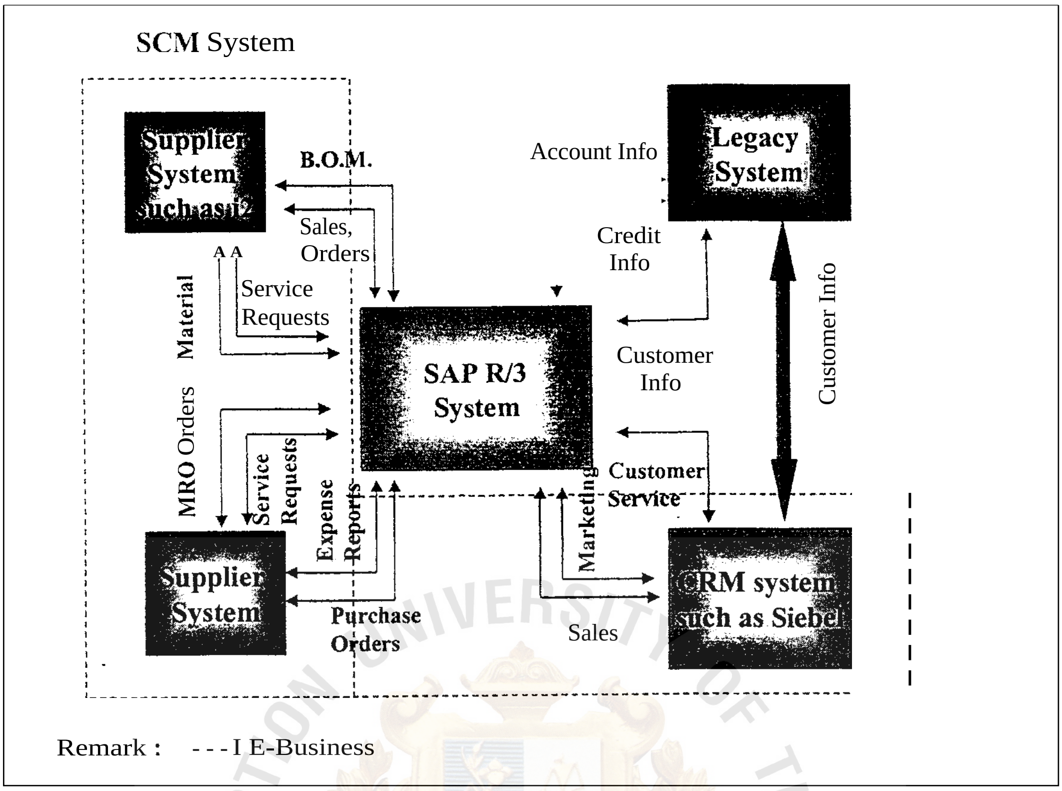


Figure 4-2 Integrated Point Model for Customized Application Integration

4.4 Flow of Customized Application Integration Prototype Program

This prototype program is simulated for whole integration between SAP R/3 system and other systems such as e-Business and Legacy system, which SAP R/3 is the key driver for this prototype. This integration includes inbound and outbound programs in SAP R/3 system by using file transfer to get incoming file and send out going file to other systems. For getting file from another system, SAP R/3 will use content in the file to generate goods movement document and transfer out into out going file to the other system. This prototype uses batch control system to generate header file before sending out the file for the design of database in SAP R/3 system will add batch number field to be the indicator of the records that was sent to other systems.

For inbound interface program, SAP R/3 will get data from incoming file, which are good receipts and good issues and posting document in SAP R/3 system and SAP R/3 system will generate good movements document and program will write report for success and fail records, which can explain as figure below:

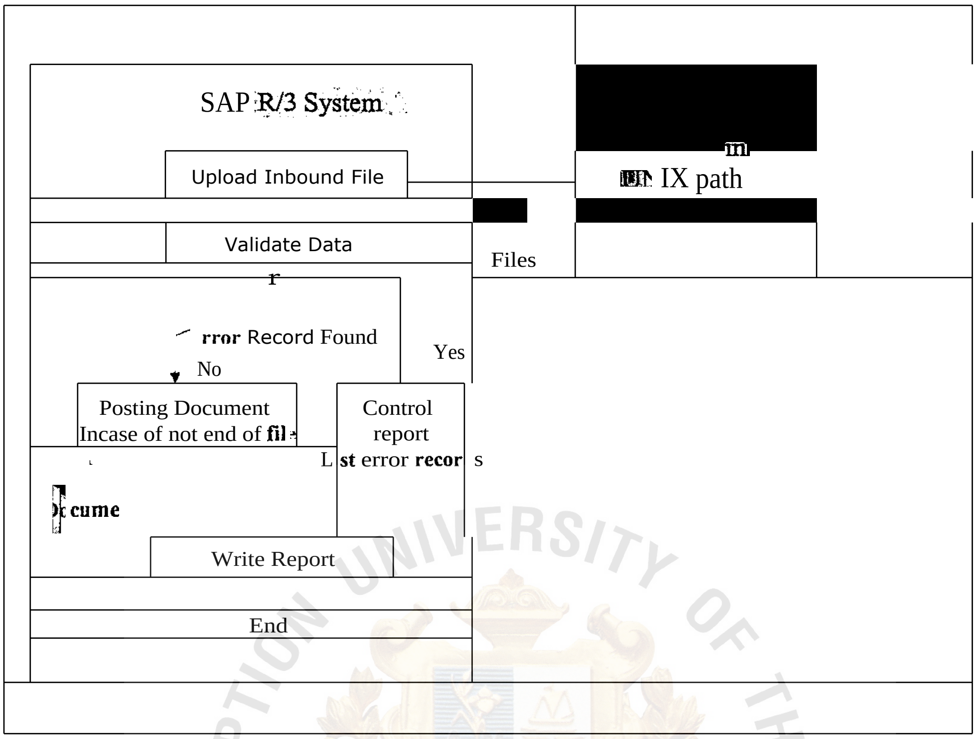


Figure 4-3 Inbound Interface flow of Customized Application Integration program

Outbound interface program of SAP R/3 will get data from SAP R/3 system for external need and send out the file to other systems by using batch control system to identify which record was sent out, which is explained in figure below:

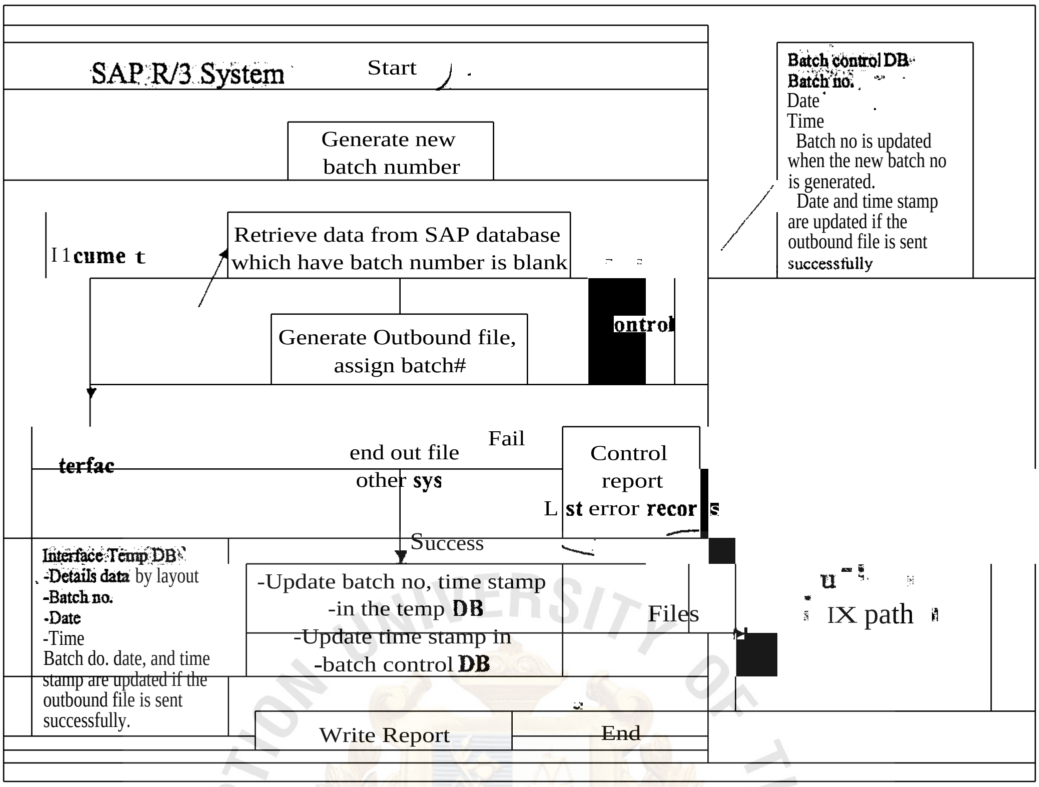

SAP R/3 System
Start
Generate new batch number
Retrieve data from SAP database which have batch number is blank
I 1cume t
Generate Outbound file, assign batch#
ontrol
terfac
end out file other sys
Fail
Control report
L st error recor s
Success
Interface Temp DB
-Details data by layout
-Batch no.
-Date
-Time
Batch do. date, and time stamp are updated if the outbound file is sent successfully.
-Update batch no, time stamp -in the temp DB
-Update time stamp in -batch control DB
Files
IX path
Write Report
End
Batch control DB
Batch no.
Date
Time
Batch no is updated when the new batch no is generated.
Date and time stamp are updated if the outbound file is sent successfully

Figure 4-4 Outbound Interface flow of Customized Application Integration program

This outbound program will generate files and retrieve all data in interface database with field batch number initially, after that will generate batch number and send file with the standard text format and after success, will update batch number, date and time into interface database and batch control database. For the next run program, this outbound program also can send the unsuccessful data from the previous batch out to the external system.

4.5 Detail design of Customized Application Integration Prototype Program

This prototype program can be separated into 2 programs, which are inbound and outbound programs.

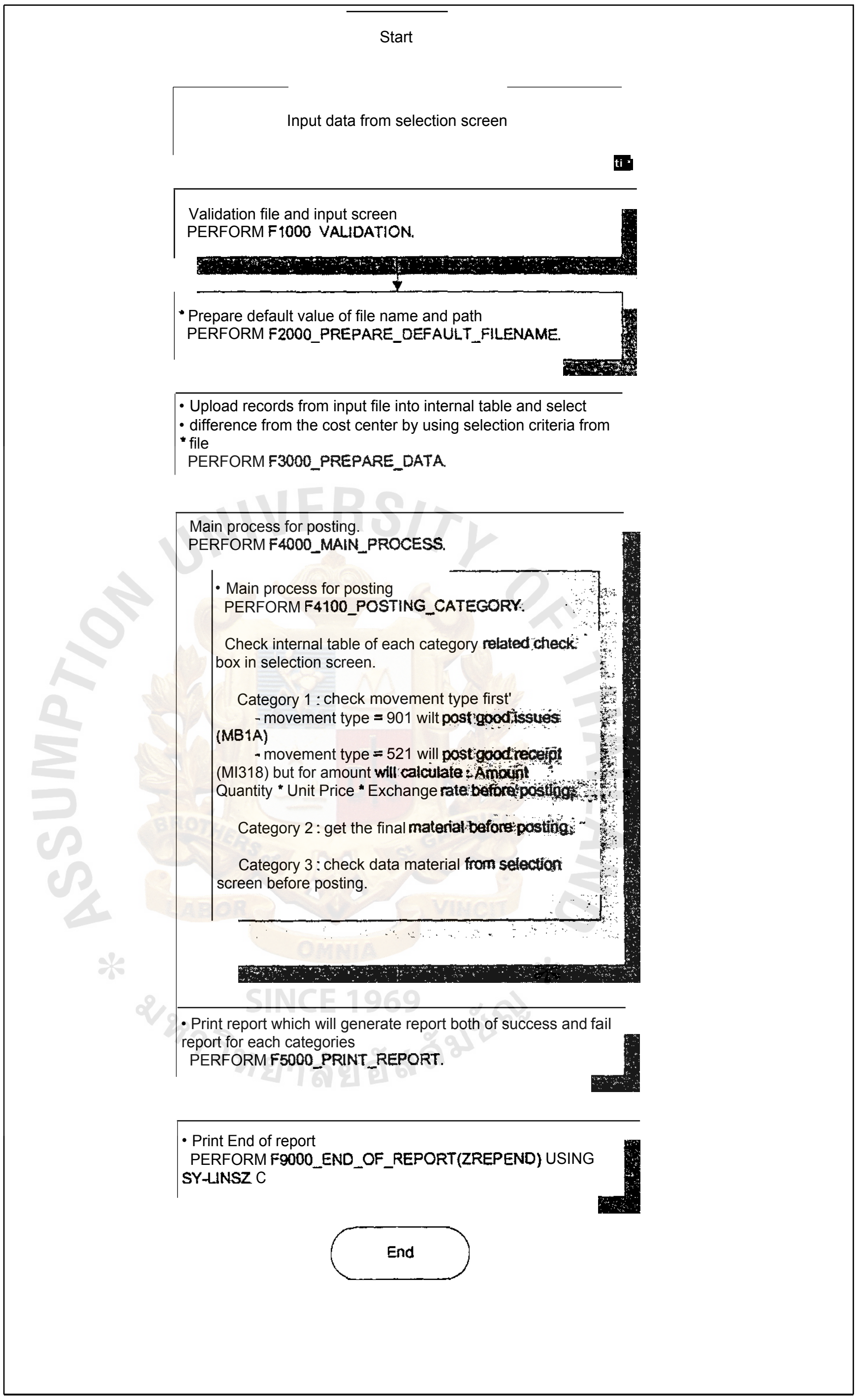
4.5.1 Inbound Program:

Inbound program is designed to upload incoming file with the movement quantities of stock into SAP R/3 system via text file. This inbound program will support only text file by using tab-delimited format. This interface program will involve three main categories, which are:

1. The first category : The interface program will read data from input file and file, calculate the amount to company code currency and post these data into SAP R/3 system. For category 1 will post into 2 BDC transactions of SAP R/3 system, separate by movement type (First 3 characters of each record)
    - Movement type 901 : will post Good issues via transaction MB1A.
    - Movement type 521 : will post Good receipt via transaction MB1B.
2. The second category : The second category is to calculate difference of debit and credit amount posted in the specified cost centers from input file and post into SAP system.

3. The last category : The third category, data from external system will be used for calculating and posting amount of utility to inventory and post into SAP R/3 system.

Inbound Technical Program Flow:



Input data from selection screen

Validation file and input screen
PERFORM F1000 VALIDATION.

* Prepare default value of file name and path
PERFORM F2000_PREPARE_DEFAULT_FILENAME.

• Upload records from input file into internal table and select
• difference from the cost center by using selection criteria from
* file
PERFORM F3000_PREPARE_DATA.

Main process for posting.
PERFORM F4000_MAIN_PROCESS.

• Main process for posting
PERFORM F4100_POSTING_CATEGORY.

Check internal table of each category related check box in selection screen.

Category 1 : check movement type first'
- movement type = 901 wilt post good issues (MB1A)
- movement type = 521 will post good receipt (MI318) but for amount will calculate : Amount Quantity * Unit Price * Exchange rate before posting.

Category 2 : get the final material before posting.

Category 3 : check data material from selection screen before posting.

• Print report which will generate report both of success and fail report for each categories
PERFORM F5000_PRINT_REPORT.

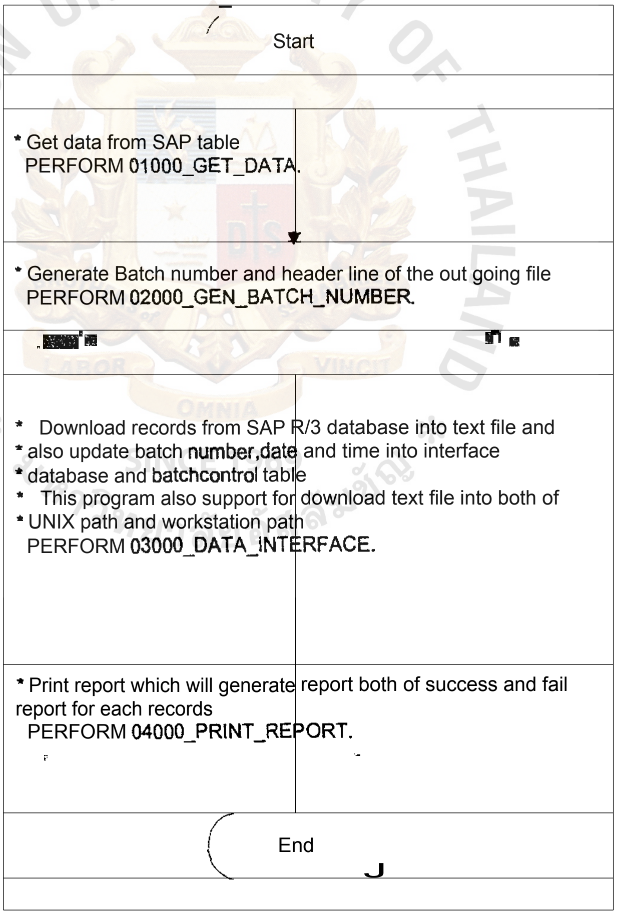
• Print End of report
PERFORM F9000_END_OF_REPORT(ZREPEND) USING SY-LINSZ C

Figure 4-5 Inbound Technical Program Flow

4.5.2 Outbound Program:

Outbound program is designed to download data from SAP R/3 system into text file and send out to external system by using batch control number to identify record transferred. Program will read only field from interface database that has field batch number initial and generate batch number into batch control database, after success transfer file program with batch number, date and time into interface database and batch control database.

Outbound Technical Program Flow:



* Get data from SAP table
  PERFORM 01000_GET_DATA.

* Generate Batch number and header line of the out going file
  PERFORM 02000_GEN_BATCH_NUMBER.

*   Download records from SAP R/3 database into text file and
* also update batch number,date and time into interface
* database and batchcontrol table
*   This program also support for download text file into both of
* UNIX path and workstation path
  PERFORM 03000_DATA_INTERFACE.

* Print report which will generate report both of success and fail report for each records
  PERFORM 04000_PRINT_REPORT.

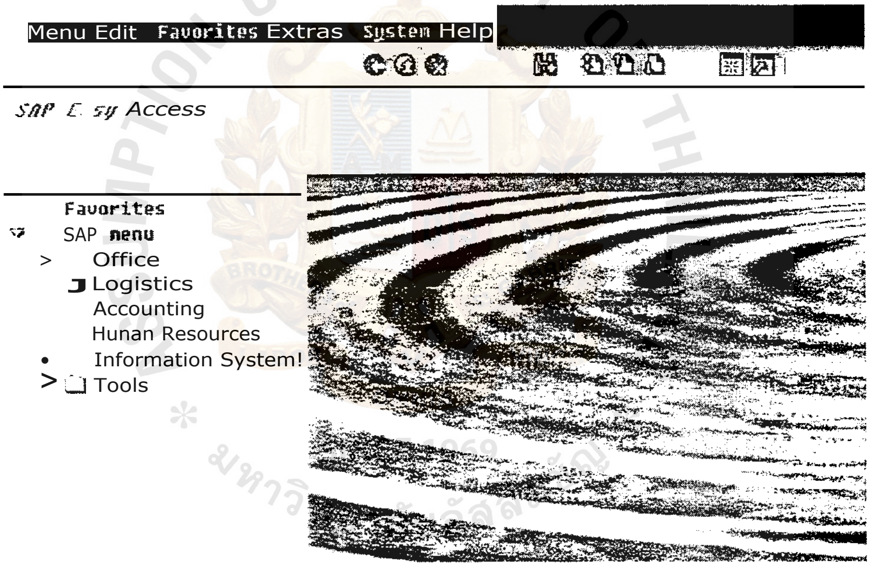
Figure 4-6 Outbound Technical Program Flow

4.6 Screen Shot of Customized Application Integration Prototype Program

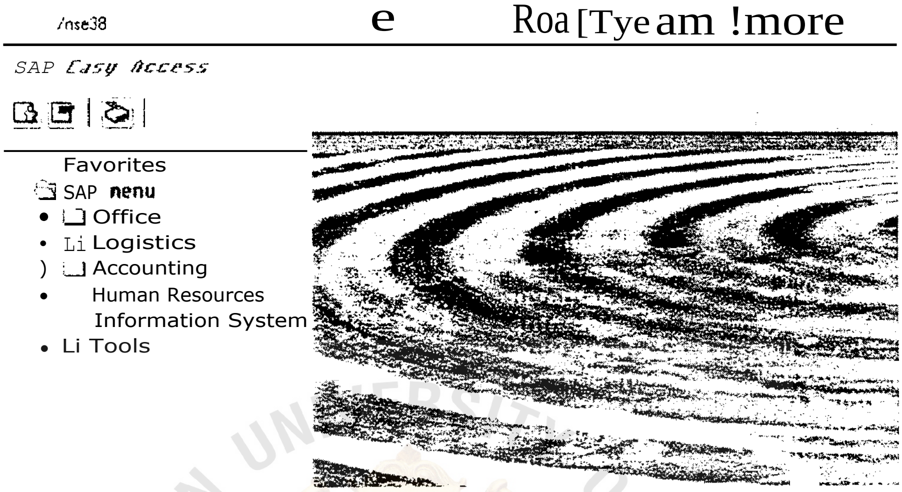
4.6.1 Screen Shot of Inbound Program:

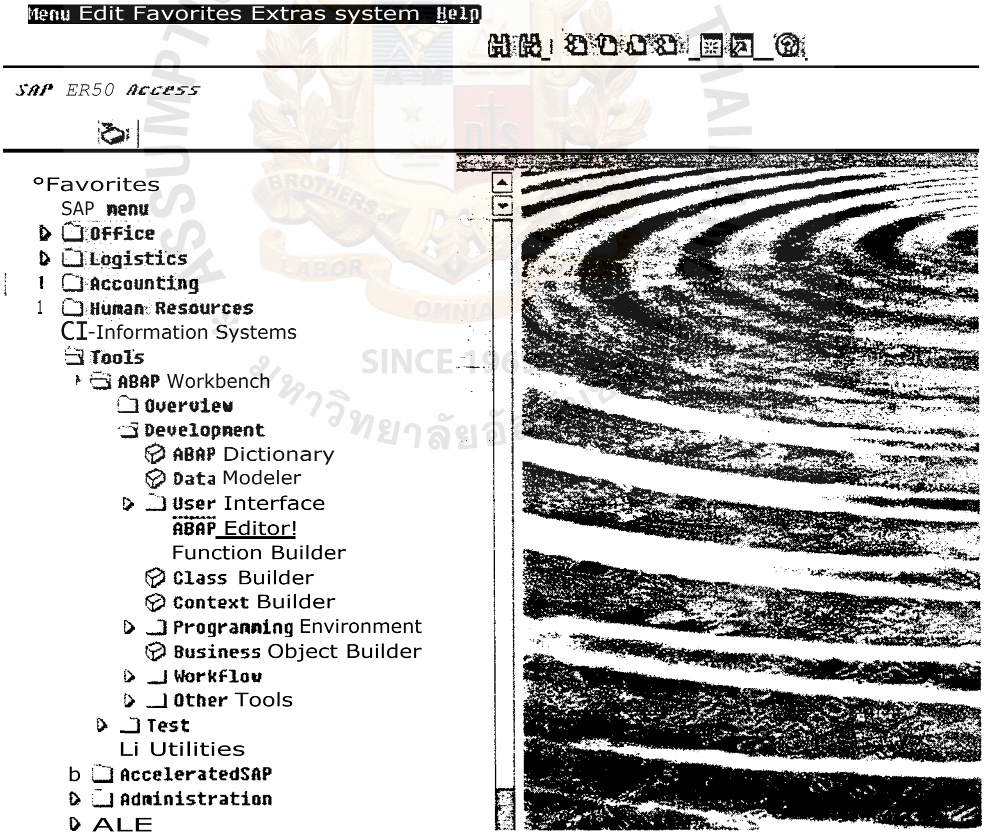
To run inbound program in SAP R/3 system, user can select via transaction SE38 or run inbound program via SAP R/3 menu path. The inbound program step can be explained as in below figure:

- Normal Screen of SAP R/3 System.

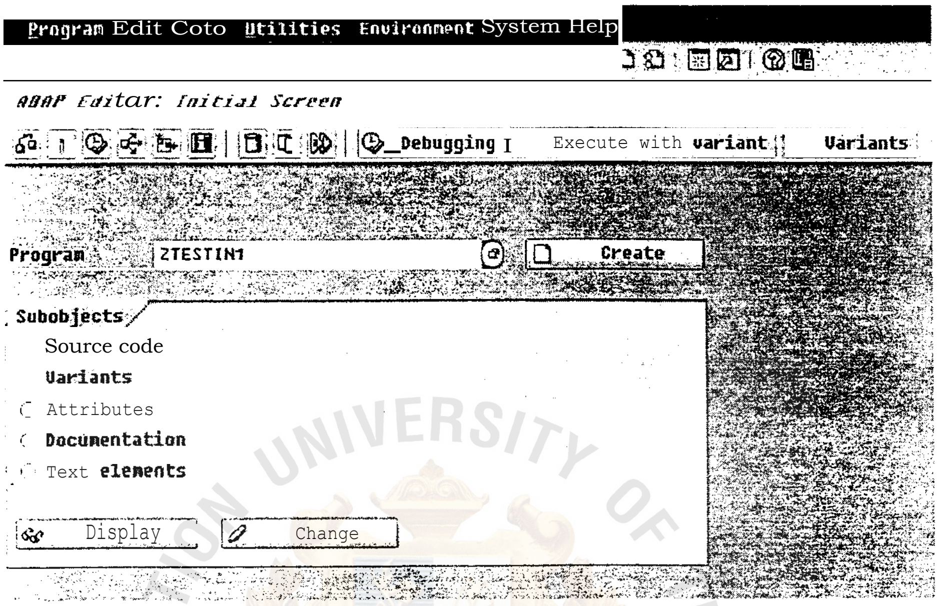


- Run program via transaction SE38 or via SAP R13 menu path.

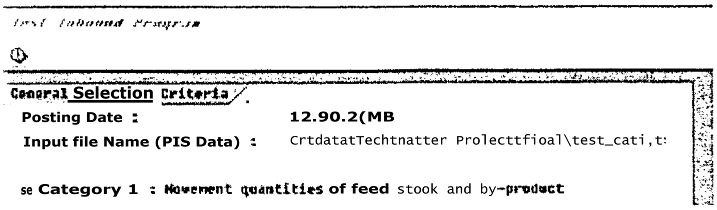




- input program name (ZTESTIN1) for Inbound program in ABAP editor and use function key (F8) to run program or click for to program.



- Run test program of category 1 : Movement quantities, Program will generate report but no posting data into SAP R/3 system.



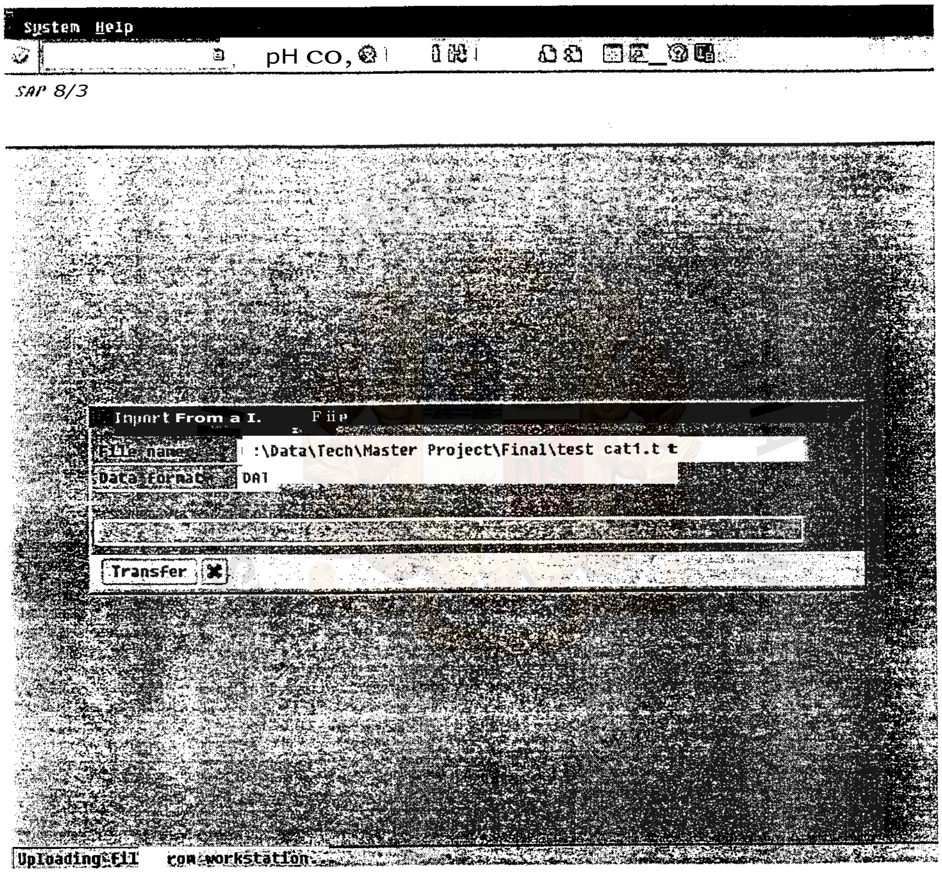
Test Inbound Program

General Selection Criteria

Posting Date : 12.90.2(MB

Input file Name (PIS Data) : CrtdatatTechtnatter Prolecttfioal\test_cati,t:

se Category 1 : Movement quantities of feed stook and by-product

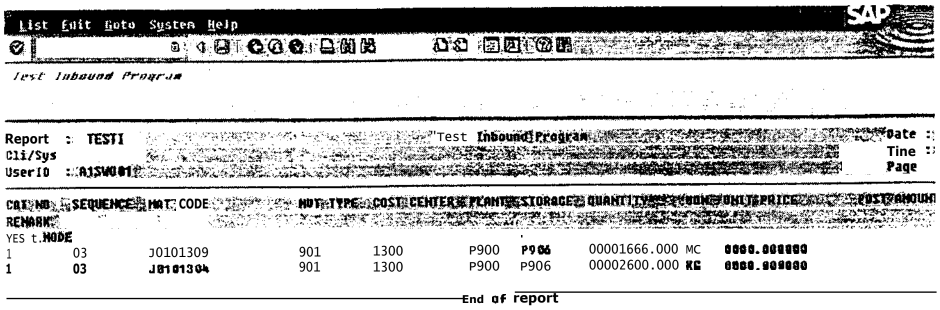


Inport From a I. File

File name : :\Data\Tech\Master Project\Final\test cat1.t t

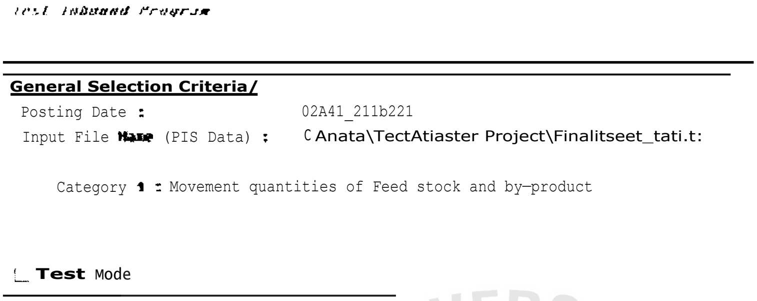
Data format DA1

Uploading fil rom workstation



```
Report  :  TESTI                         "Test Inbound Program                                   Date :
Cli/Sys                                                                                          Tine :
UserID  :  A1SW001                                                                               Page

CAT NO  SEQUENCE  MAT CODE          MOT TYPE  COST CENTER  PLANT  STORAGE  QUANTITY       UOM  UNIT PRICE       COST AMOUNT
REMARK
YES t.MODE
1       03        J0101309          901       1300         P900   P906     00001666.000   MC   0000.000000
1       03        J0101304          901       1300         P900   P906     00002600.000   KG   0000.000000
                                             End of report
```

- Run post program of category 1 : Movement quantities, Program will generate report and posting data into SAP R/3 system. (Unmark Test Check box)



Test Inbound Program

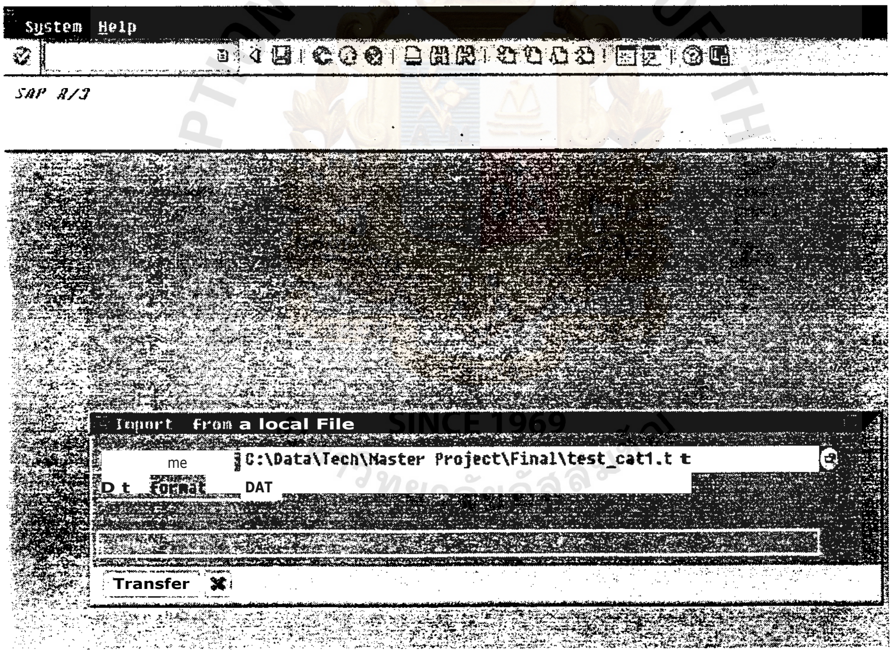
General Selection Criteria/

Posting Date : 02A41_211b221

Input File Name (PIS Data) : C Anata\TectAtiaster Project\Finalitseet_tati.t:

Category 1 : Movement quantities of Feed stock and by—product

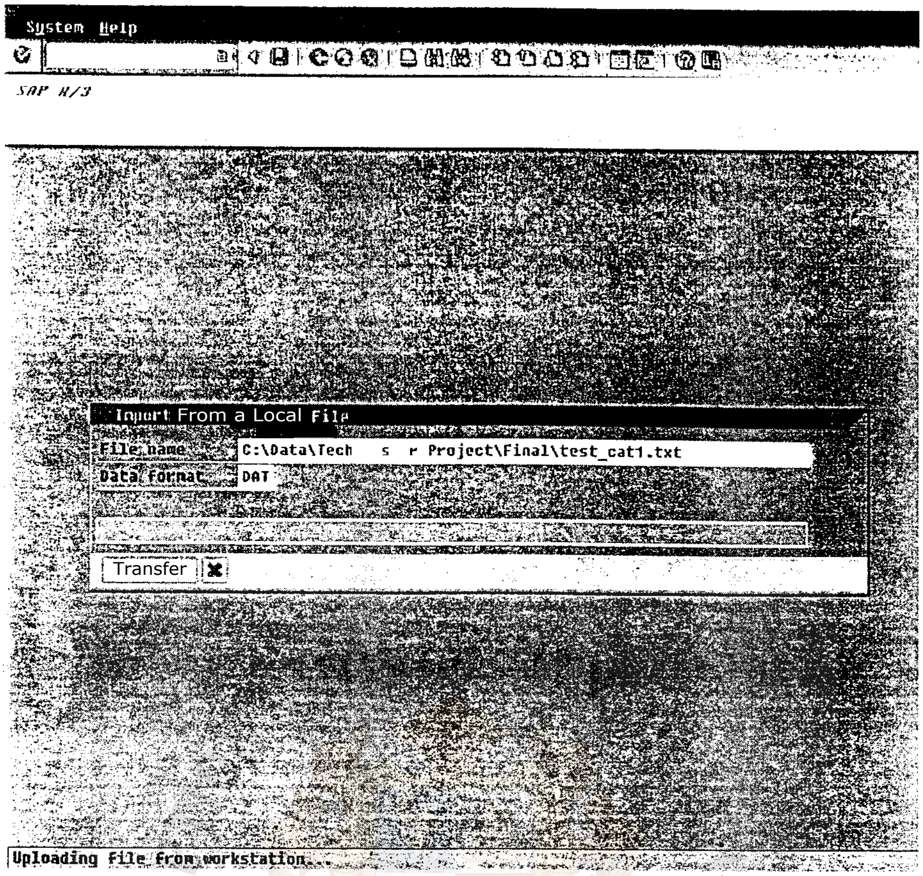
□ **Test** Mode



Import from a local File

me C:\Data\Tech\Master Project\Final\test_cat1.t t

D t Format DAT

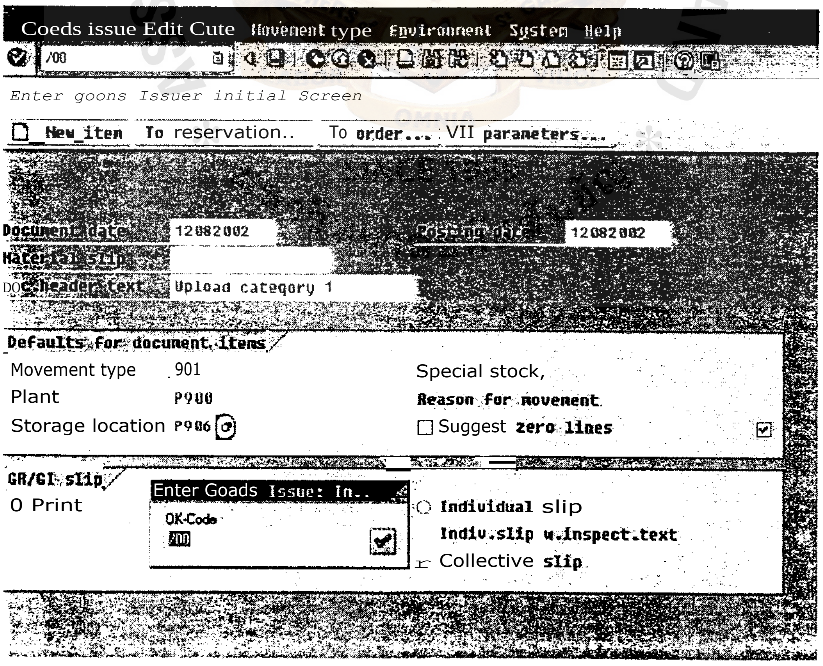


Import From a Local File

File name C:\Data\Tech s r Project\Final\test_cat1.txt

Data format DAT

Uploading file from workstation...


Coeds issue Edit Cute Movement type Environment System Help

Enter goons Issuer initial Screen
New item To reservation.. To order... VII parameters...

Document date 12082002 Posting date 12082002

Material slip

Doc.header text Upload category 1

Defaults for document items

Movement type 901

Plant P900

Storage location P906

Special stock,

Reason for movement

□ Suggest zero lines

GR/GI slip

0 Print

Enter Goads Issue: In..

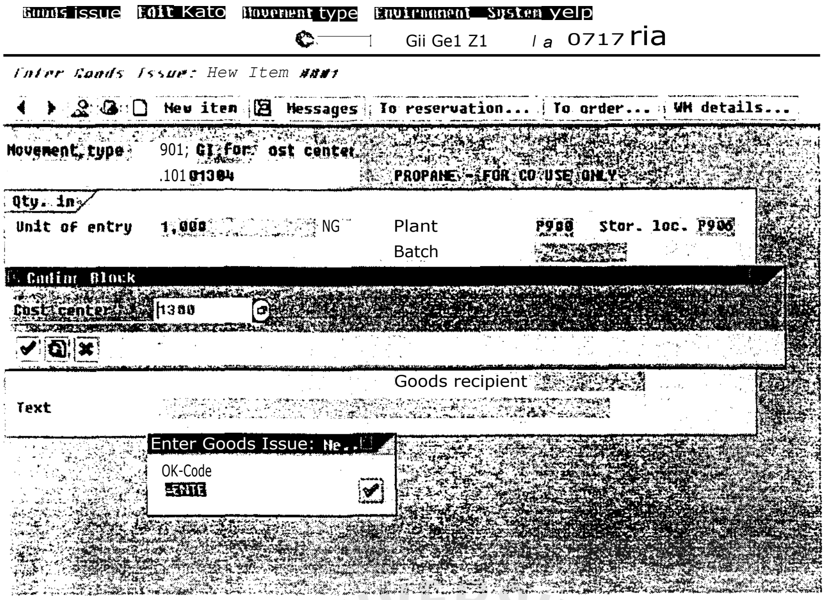
OK-Code

/00

○ Individual slip

Indiv.slip w.inspect.text

Collective slip



Goods issue Edit Kato Movement type Environment System yelp

Gii Ge1 Z1 / a 0717 ria

Enter Goods Issue: Hew Item 0001

New item | Messages | To reservation... | To order... | WM details...

Movement type 901; GI for ost center

.101 01304 PROPANE - FOR CO USE ONLY

Qty. in

Unit of entry 1,000 NG Plant P900 Stor. loc. P906

Batch

Coding Block

Cost center 1300

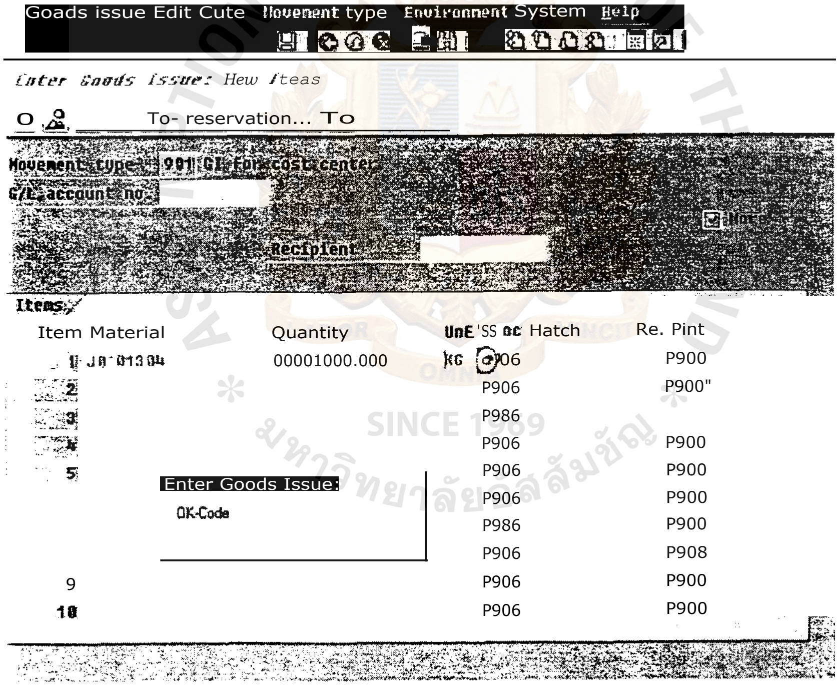
Goods recipient

Text

Enter Goods Issue: Ne...

OK-Code

ENTE



Goads issue Edit Cute Movement type Environment System Help

Enter Goods Issue: Hew Items

To- reservation... To

Movement type 901 GI for cost center

G/L account no.

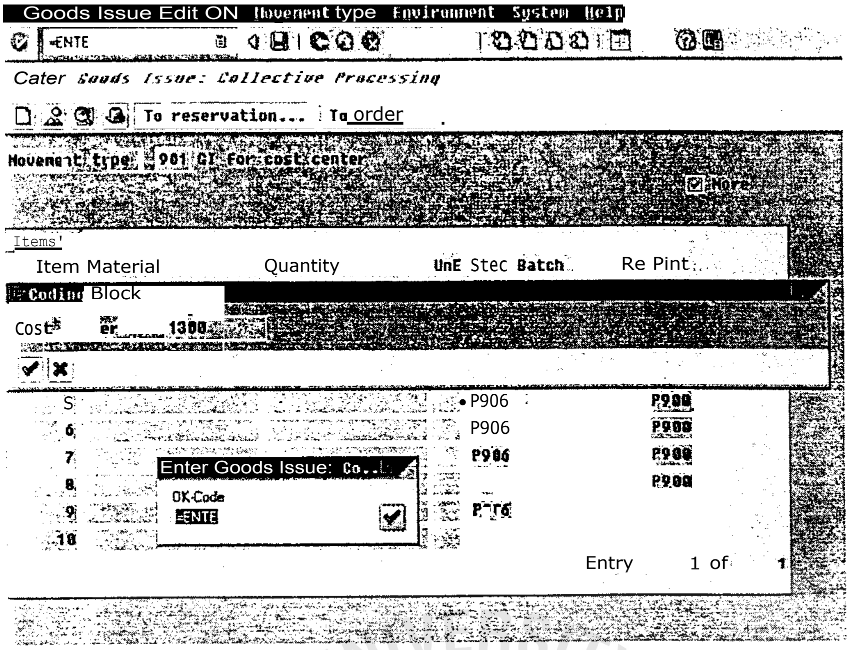
Recipient

Items

Item	Material	Quantity	UnE 'SS oc	Hatch	Re. Pint
1	JA 01304	00001000.000	KG	P906	P900
2				P906	P900"
3				P986	
4				P906	P900
5				P906	P900
				P906	P900
				P986	P900
				P906	P908
9				P906	P900
10				P906	P900

Enter Goods Issue:

OK-Code



=ENTE

Cater Goods Issue: Collective Processing

Movement type 901 GI for cost center

Item	Material	Quantity	UnE	Stec	Batch	Re Pint
S				P906		P900
6				P906		P900
7				P906		P900
8						P900
9				P906		
10						

Coding Block

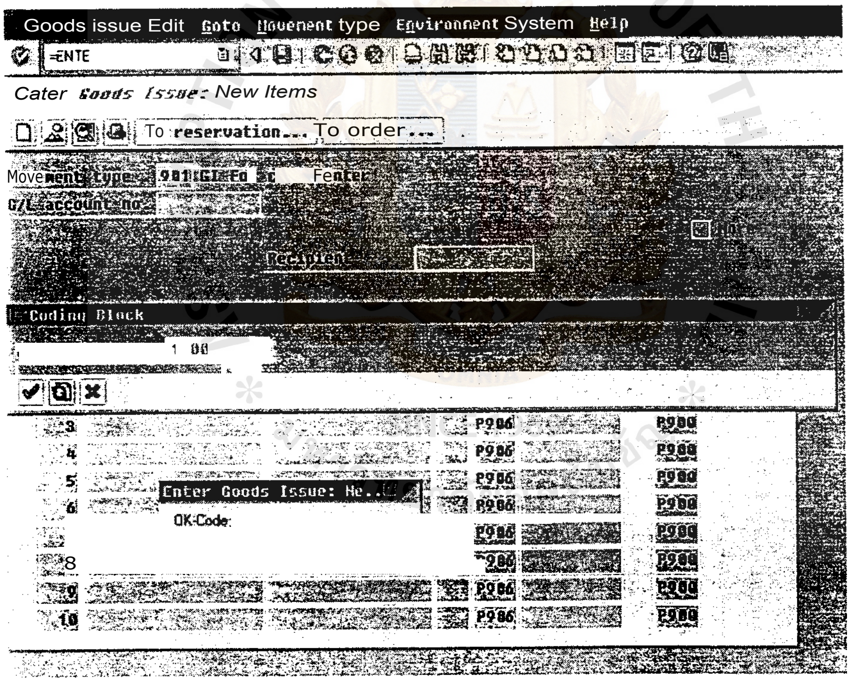
Cost center 1300

Enter Goods Issue: Co...

OK-Code

=ENTE

Entry 1 of 1



=ENTE

Cater Goods Issue: New Items

Movement type 901 GI Fo cost center

G/L account no

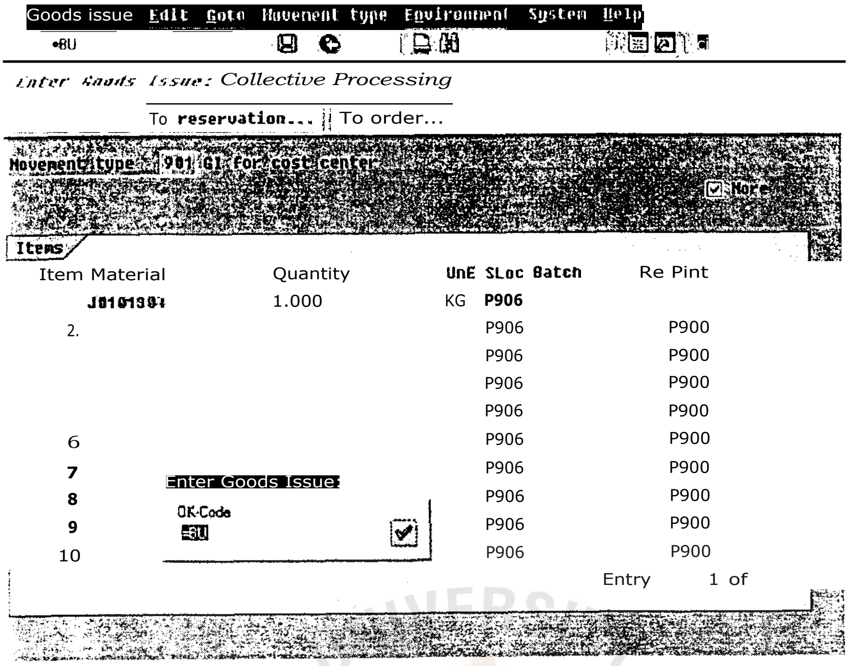
Coding Block

1 00

Item		Stec		Pint
3		P906		P900
4		P906		P900
5		P906		P900
6		P906		P900
		P906		P900
8		P906		P900
9		P906		P900
10		P906		P900

Enter Goods Issue: Ne...

OK-Code:



Enter Goods Issue: Collective Processing

To reservation... To order...

Movement type 901 GI for cost center

☑ More

Items

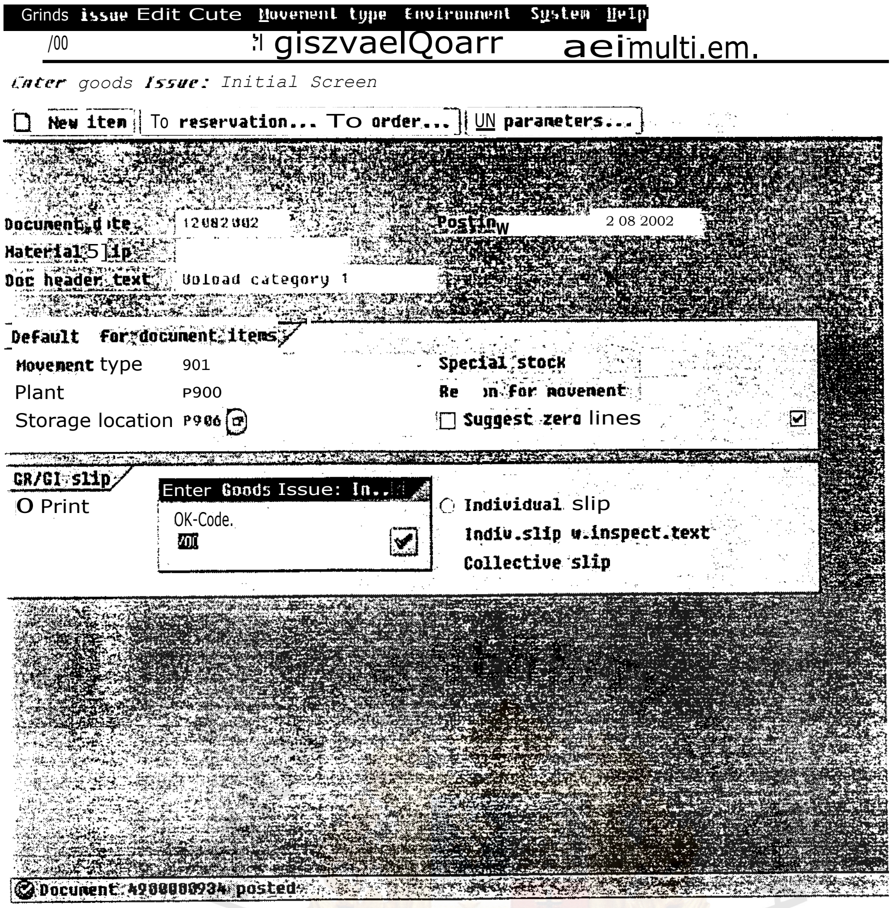
Item	Material	Quantity	UnE	SLoc	Batch	Re Pint
	J0101307	1.000	KG	P906		
2.				P906		P900
				P906		P900
				P906		P900
				P906		P900
6				P906		P900
7				P906		P900
8				P906		P900
9				P906		P900
10				P906		P900

Enter Goods Issue:

OK-Code
=BU

☑

Entry 1 of


Grinds issue Edit Cute Movement type Environment System Help
/00
giszvaelQoarr aeimulti.em.
New item
To reservation...
To order...
UN parameters...

Enter goods Issue: Initial Screen

Document date 12082002
Posting 2 08 2002
Material Slip
Doc header text Upload category 1

Default for document items

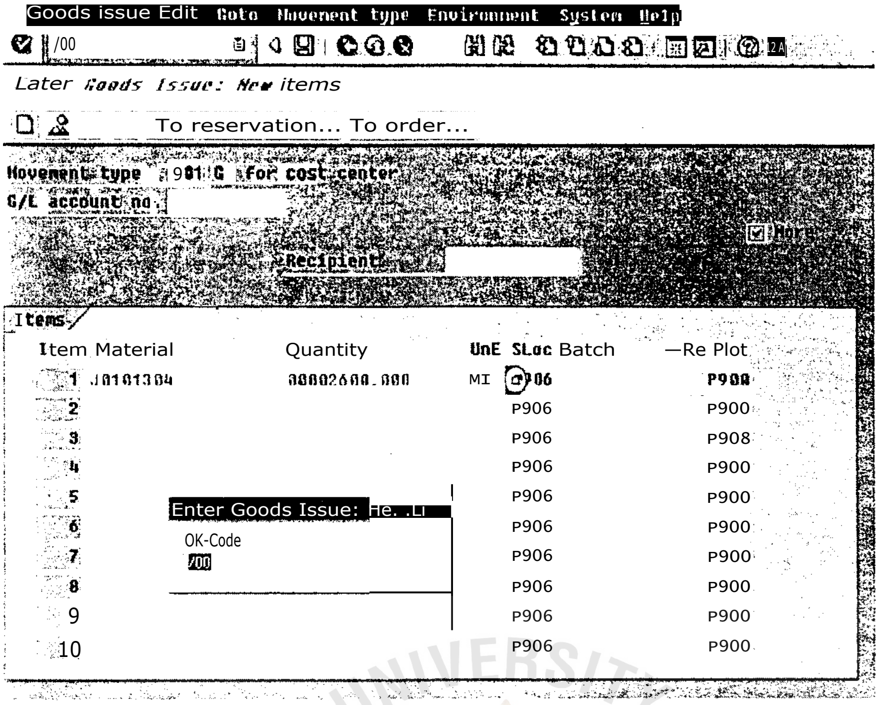
Movement type 901
Plant P900
Storage location P906
Special stock
Reason for movement
□ Suggest zero lines ☑

GR/GI slip
○ Print

Enter Goods Issue: In...
OK-Code.
/00 ☑

○ Individual slip
Indiv.slip w.inspect.text
Collective slip

Document 4900000934 posted



/00

Later Goods Issue: New items

To reservation... To order...

Movement type 901 G for cost center

G/L account no.

☑ More

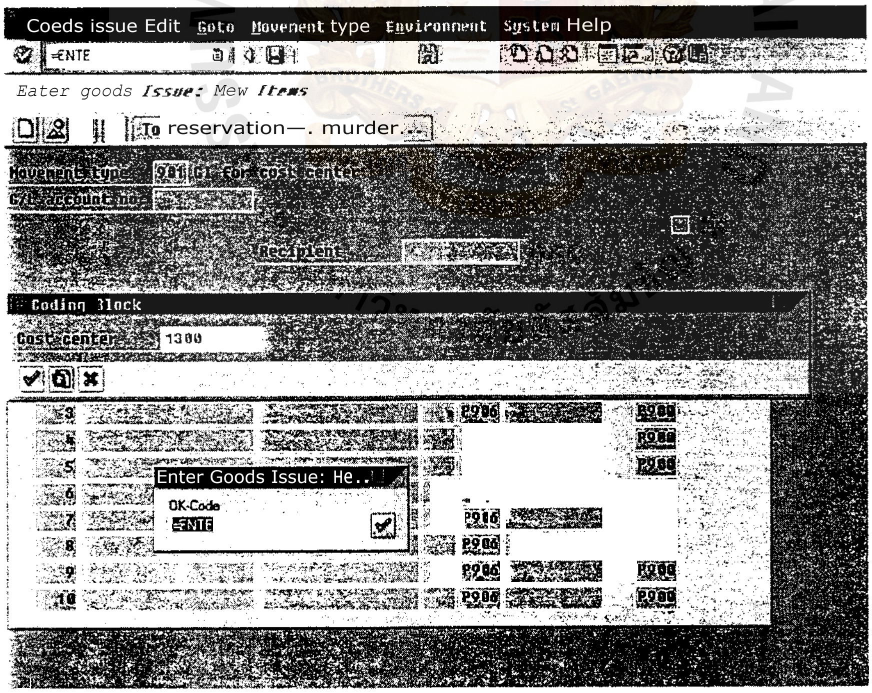
Recipient

Items

Item	Material	Quantity	UnE	SLoc	Batch	Re	Plot
1	J0101304	00002600.000	MI	P906			P900
2				P906			P900
3				P906			P908
4				P906			P900
5				P906			P900
6				P906			P900
7				P906			P900
8				P906			P900
9				P906			P900
10				P906			P900

Enter Goods Issue: He. .Li

OK-Code
/00



=ENTE

Eater goods Issue: Mew Items

To reservation—. murder...

Movement type 901 G1 for cost center

G/L account no.

Recipient

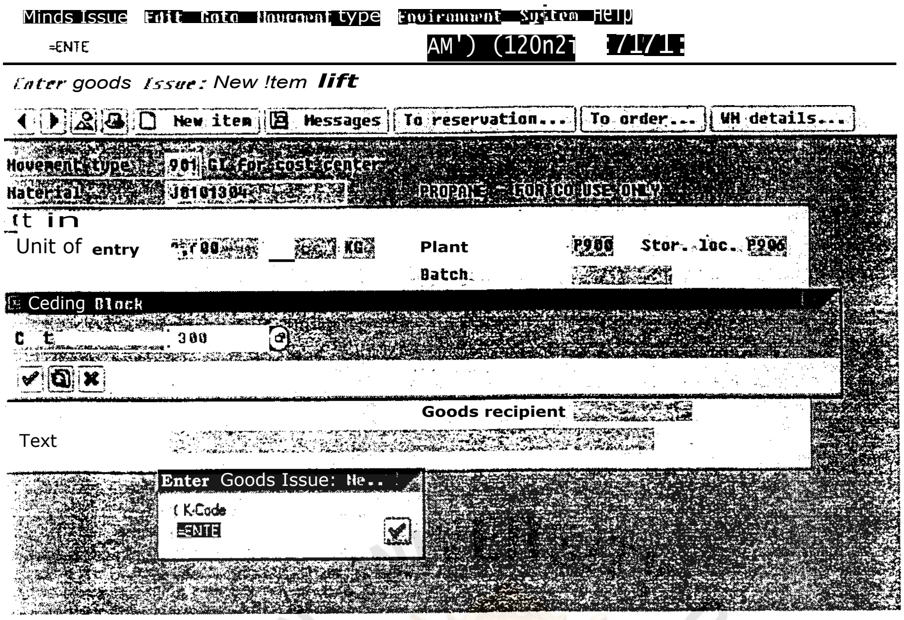
Coding Block

Cost center 1300

3	[illegible]	[illegible]	P906	[illegible]	P900
4	[illegible]	[illegible]			P900
5	[illegible]	[illegible]			P900
6	[illegible]				
7	[illegible]		P916	[illegible]	
8	[illegible]		P906		
9	[illegible]	[illegible]	P906	[illegible]	P900
10	[illegible]	[illegible]	P906	[illegible]	P900

Enter Goods Issue: He..

OK-Code
=ENTE ✓



Enter goods Issue: New Item ***lift***

Movement type 901 GI for cost center

Material J0101304 PROPANE - FOR CO USE ONLY

(t in

Unit of entry KG Plant P900 Stor. loc. P906

Batch

Ceding Block

C t 300

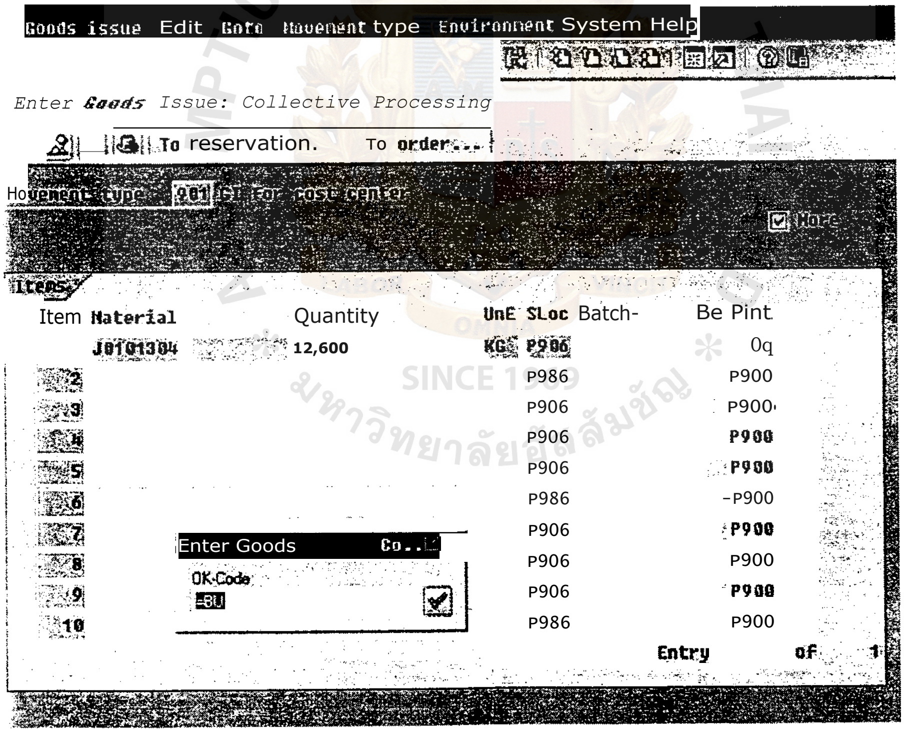
Goods recipient

Text

Enter Goods Issue: Ne..

(K-Code

=ENTE



Enter Goods Issue: Collective Processing

Movement type 901 GI for cost center

☑ More

Items

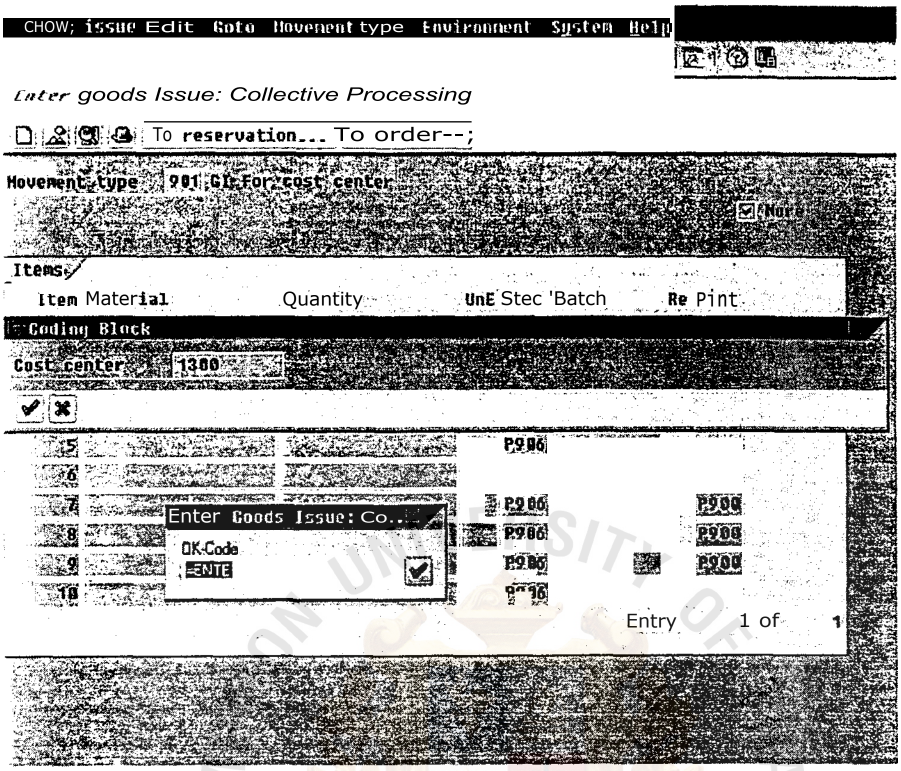
Item	Material	Quantity	UnE	SLoc	Batch-	Be Pint
	J0101304	12,600	KG	P906		0q
2				P986		P900
3				P906		P900
4				P906		P900
5				P906		P900
6				P986		-P900
7				P906		P900
8				P906		P900
9				P906		P900
10				P986		P900

Entry of 1

Enter Goods Co..

OK-Code

=BU


CHOW; issue Edit Goto Hovenent type Environnent System Help
To reservation... To order--;
Items
Item Material Quantity UnE Stec 'Batch Re Pint
Coding Block
Enter Goods Issue: Co...
OK-Code
Entry 1 of 1

Enter goods Issue: Collective Processing

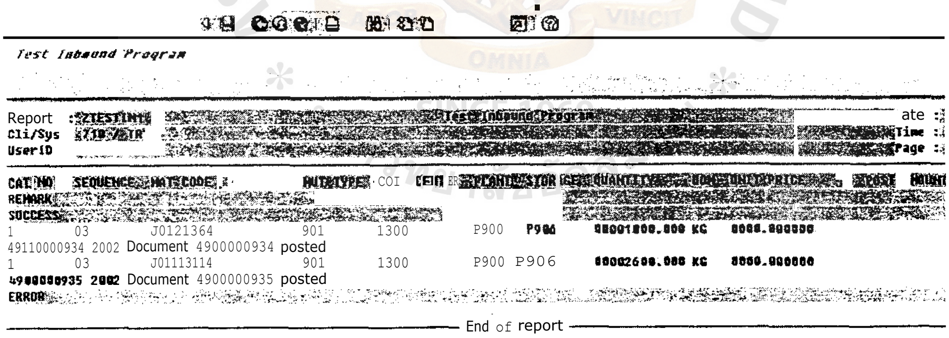
Hovenent type 901 GI For cost center

☑ Nore

Cost center 1300

OK-Code ENTE

Item	Material	Quantity	UnE	Stec	'Batch	Re	Pint
5				P906			
6							
7				P906			P900
8				P906			P900
9				P906			P900
10				P906			



Test Inbound Program

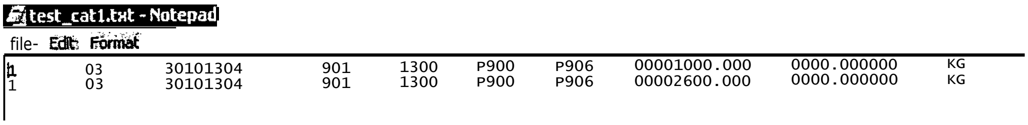
```
Report   :ZTESTINTE            Test Inbound Program                 ate :
Cli/Sys  :710 /TR                                                   Time :
UserID                                                              Page :

CAT NO  SEQUENCE  MATCODE           MVTTYPE COI   CEN ER PLANT STOR GE QUANTITY  UOM UNITPRICE  COST AMOUNT
REMARK
SUCCESS
1        03       J0121364          901     1300      P900  P906   00001000.000 KG   0000.000000
49110000934 2002 Document 4900000934 posted
1        03       J01113114         901     1300      P900  P906   00002600.000 KG   0000.000000
4900000935 2002 Document 4900000935 posted
ERROR
------------------------------ End of report ------------------------------
```

- Example of incoming file for Run post program of category 1 : Movement quantities.

```
1         03         J0101304         901 1300 P900 P906 00001000.000
0000.000000 KG
1         03         J0101304         901 1300 P900 P906 00002600.000
0000.000000 KG
```

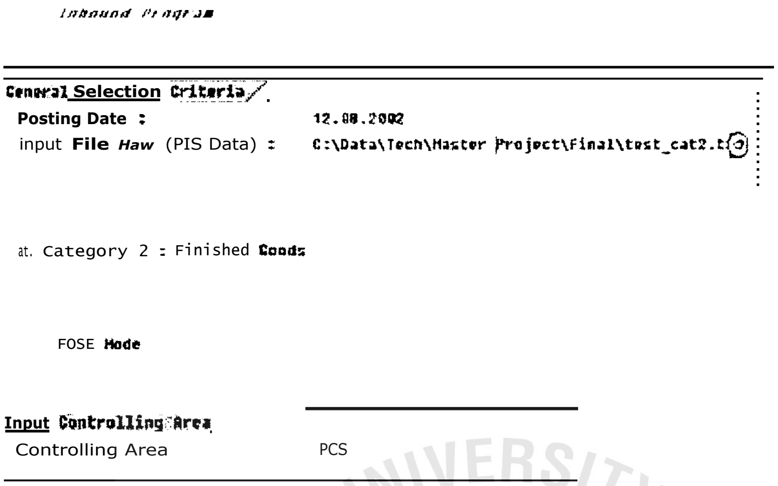
Format file :

1	: is category number 1
03	: is sequence of file
J0101304	: is Material number
901	: is Movement Type
1300	: is Cost Center
P900	: is Storage Location
P906	: is Plant
1000	: is quantity
KG	: is Unit of measurement



```
1        03       30101304        901      1300     P900     P906     00001000.000     0000.000000     KG
1        03       30101304        901      1300     P900     P906     00002600.000     0000.000000     KG
```

- Run test program of category 2 : Finished Goods, Program will generate report but no posting of data into SAP R/3 system.



Inbound Program

General Selection Criteria

Posting Date : 12.08.2002

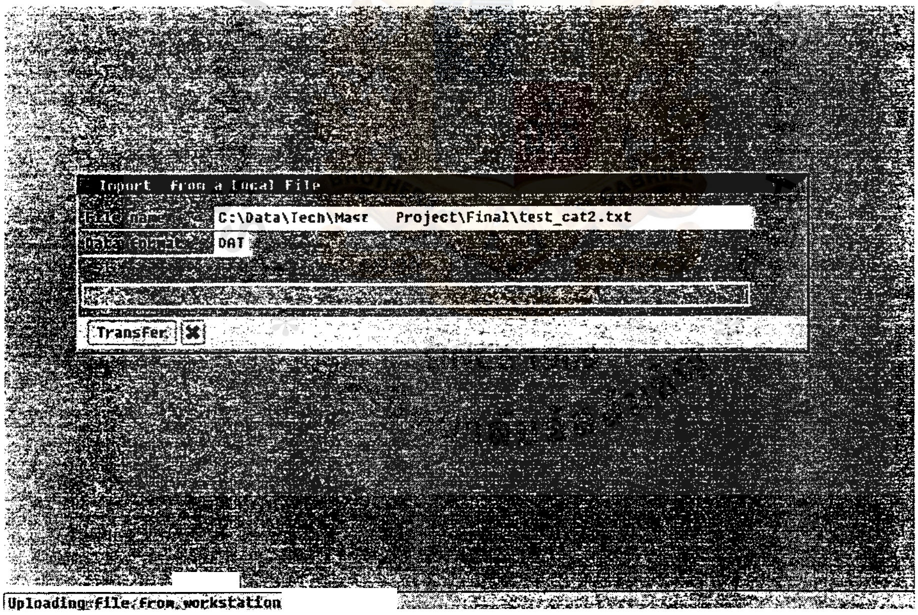
input File Haw (PIS Data) : C:\Data\Tech\Master Project\Final\test_cat2.t

at. Category 2 : Finished Goods

FOSE Mode

Input Controlling Area

Controlling Area PCS

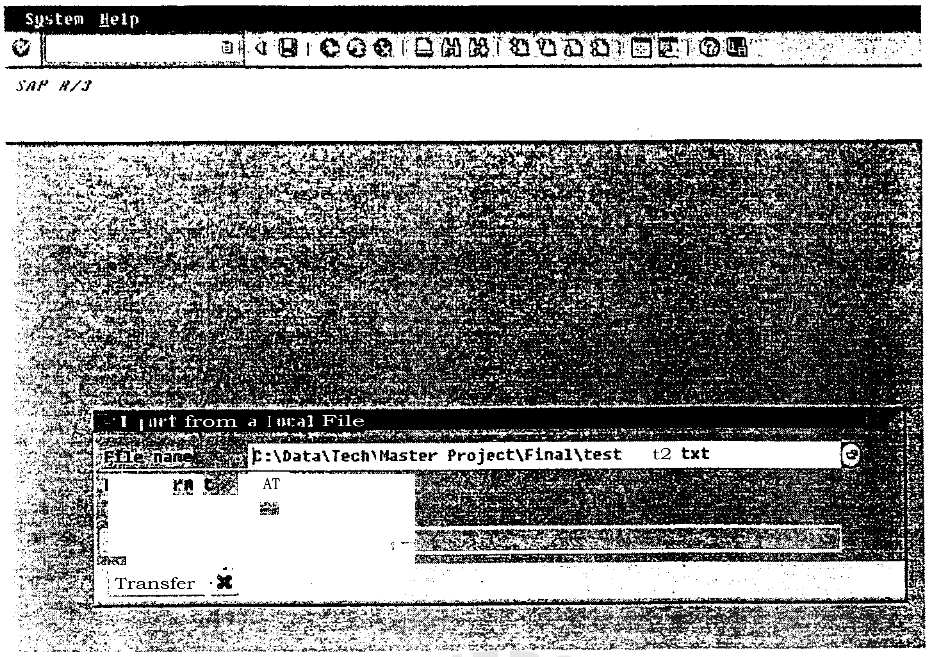


Import from a Local File

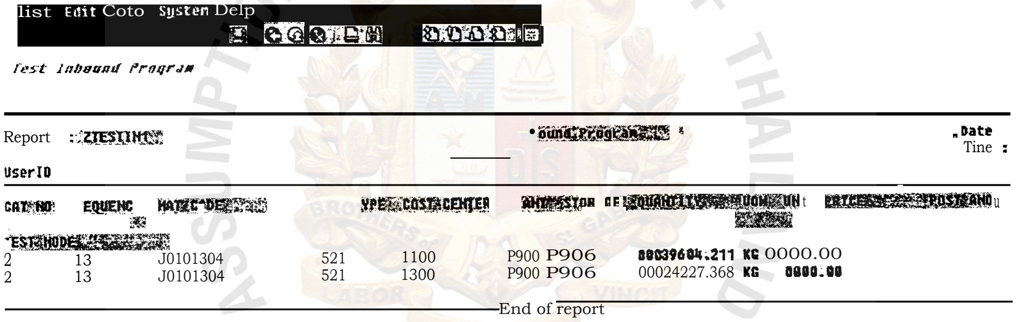
File name C:\Data\Tech\Mast Project\Final\test_cat2.txt

Data format DAT

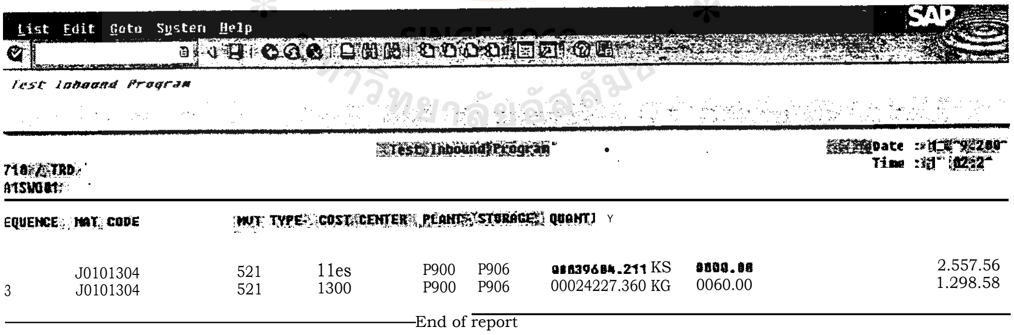
Uploading file from workstation



```
I port from a local File
File name  C:\Data\Tech\Master Project\Final\test  t2 txt
```

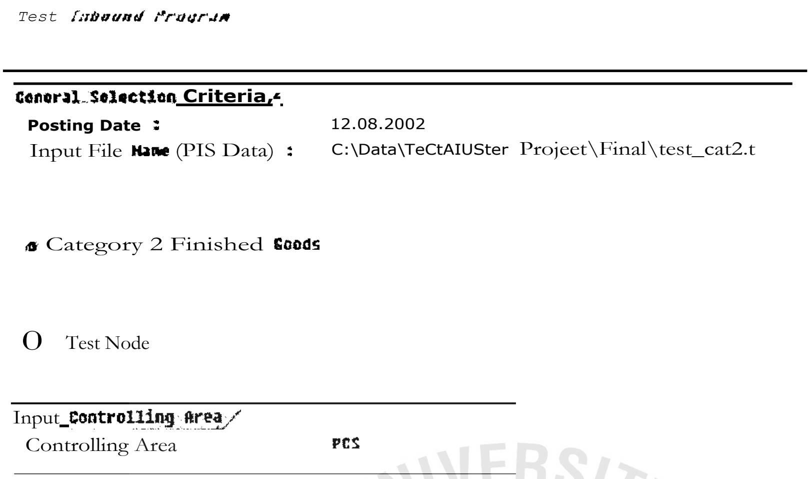


```
Report  :ZTESTIN1                   •ound Program                     .Date
                                                                      Tine :
UserID
CAT NO   EQUENC  MATL DE          VPE COSTACENTER  AHT STOR CE QUANTITY UOM UNt  PRICE  POSIEAHOu
EST MODEL
2        13      J0101304         521    1100      P900 P906   00039604.211 KG 0000.00
2        13      J0101304         521    1300      P900 P906   00024227.368 KG   0000.00
                                        End of report
```



```
                              Test Inbound Program                        Date :1 9 280
718 TRD                                                                   Time :1  02:2
A1SV001
EQUENCE  MAT CODE          MVT TYPE  COST CENTER  PLANT  STORAGE  QUANTI Y

         J0101304          521    11es     P900  P906  00039604.211 KS   0000.00        2.557.56
3        J0101304          521    1300     P900  P906  00024227.360 KG   0060.00        1.298.58
                                   End of report
```

- Run post program of category 2 : Finished Goods, Program will generate report and post data into SAP R/3 system. (Unmark Test Check box)



Test Inbound Program

Conoral Selection Criteria

Posting Date : 12.08.2002

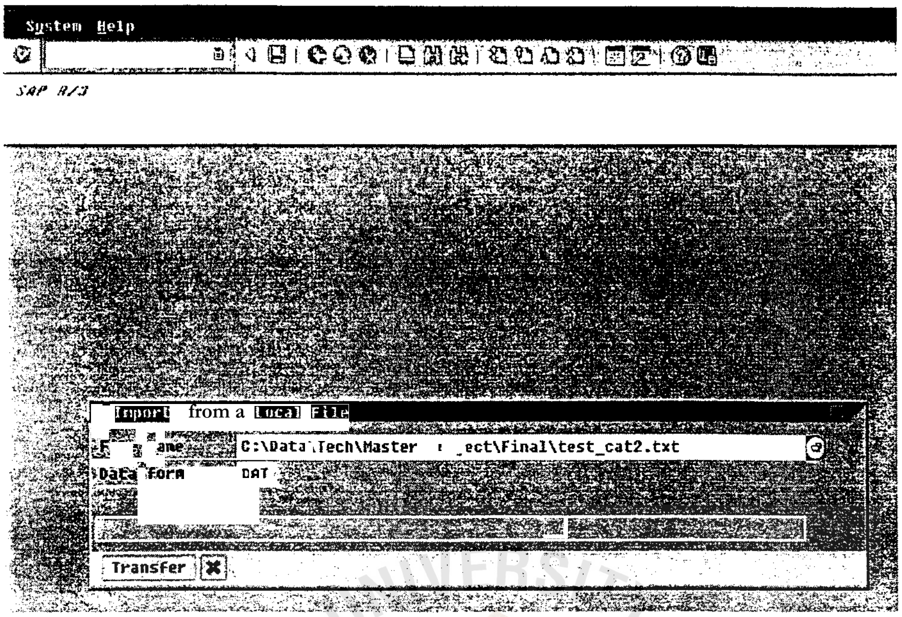
Input File Name (PIS Data) : C:\Data\TeCtAIUSter Projeet\Final\test_cat2.t

● Category 2 Finished Goods

○ Test Node

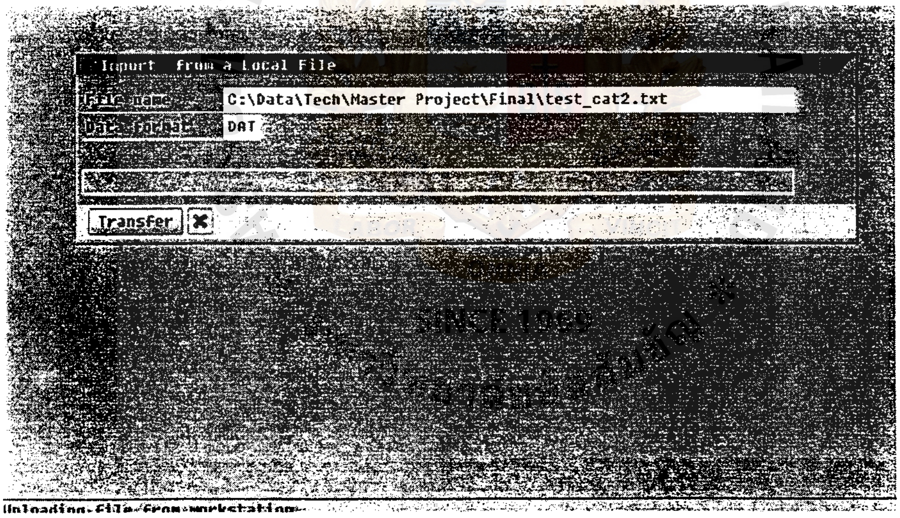
Input Controlling Area

Controlling Area PCS



File name: C:\Data\Tech\Master _ect\Final\test_cat2.txt

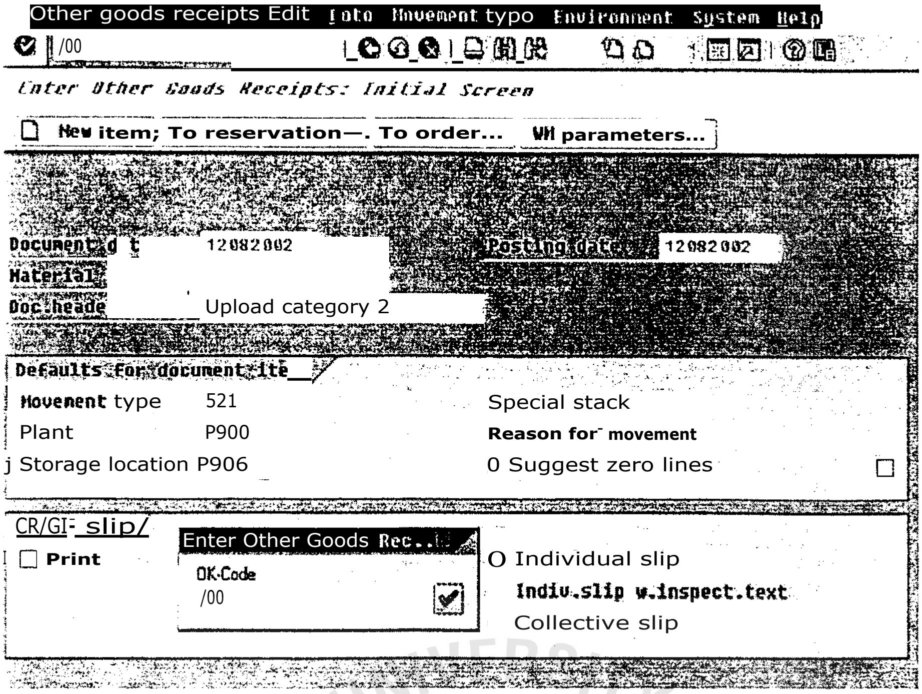
Data format: DAT



File name: C:\Data\Tech\Master Project\Final\test_cat2.txt

Data format: DAT

Uploading file from workstation



Other goods receipts Edit Goto Movement typo Environment System Help

/00

Enter Other Goods Receipts: Initial Screen

New item; To reservation—. To order... WM parameters...

Document d t 12082002 Posting date 12082002

Material

Doc.heade Upload category 2

Defaults for document ite

Movement type 521

Plant P900

Storage location P906

Special stock

Reason for movement

Suggest zero lines □

CR/GI slip

□ Print

Enter Other Goods Rec...

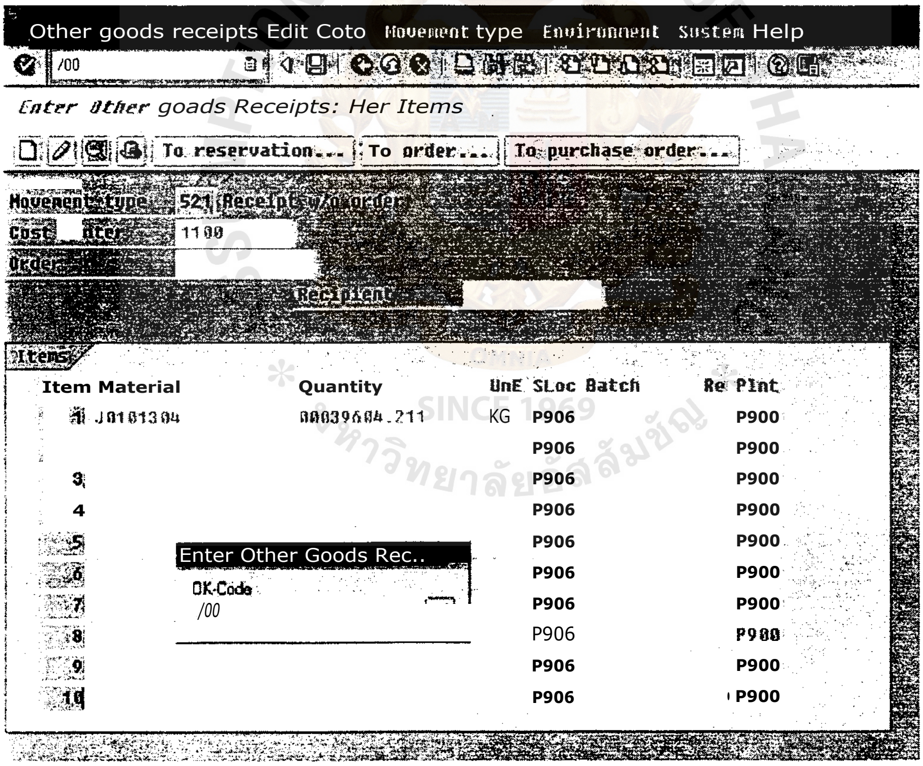
OK-Code

/00

○ Individual slip

Indiv.slip w.inspect.text

Collective slip



Other goods receipts Edit Coto Movement type Environment System Help

/00

Enter Other goads Receipts: Her Items

To reservation... To order... To purchase order...

Movement type 521 Receipt w/o order

Cost center 1100

Order

Recipient

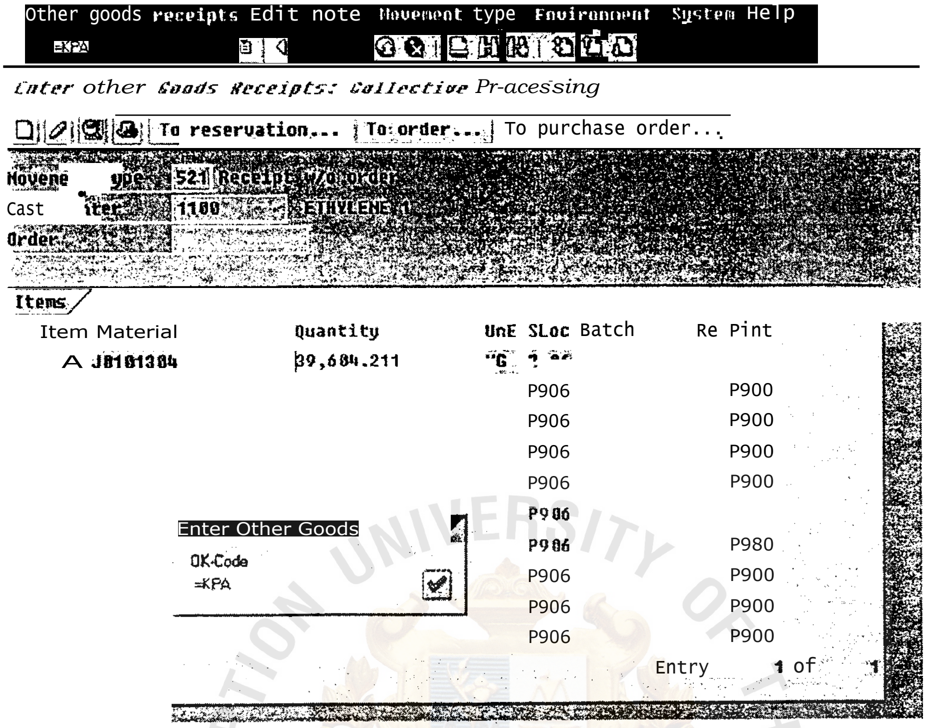
Items

Item	Material	Quantity	UnE	SLoc	Batch	Re	Plnt
1	J0101304	00039604.211	KG	P906			P900
				P906			P900
3				P906			P900
4				P906			P900
5				P906			P900
6				P906			P900
7				P906			P900
8				P906			P900
9				P906			P900
10				P906			P900

Enter Other Goods Rec..

OK-Code

/00



Enter other Goods Receipts: Collective Pr-acessing

Movene type 521 Receipt w/o order

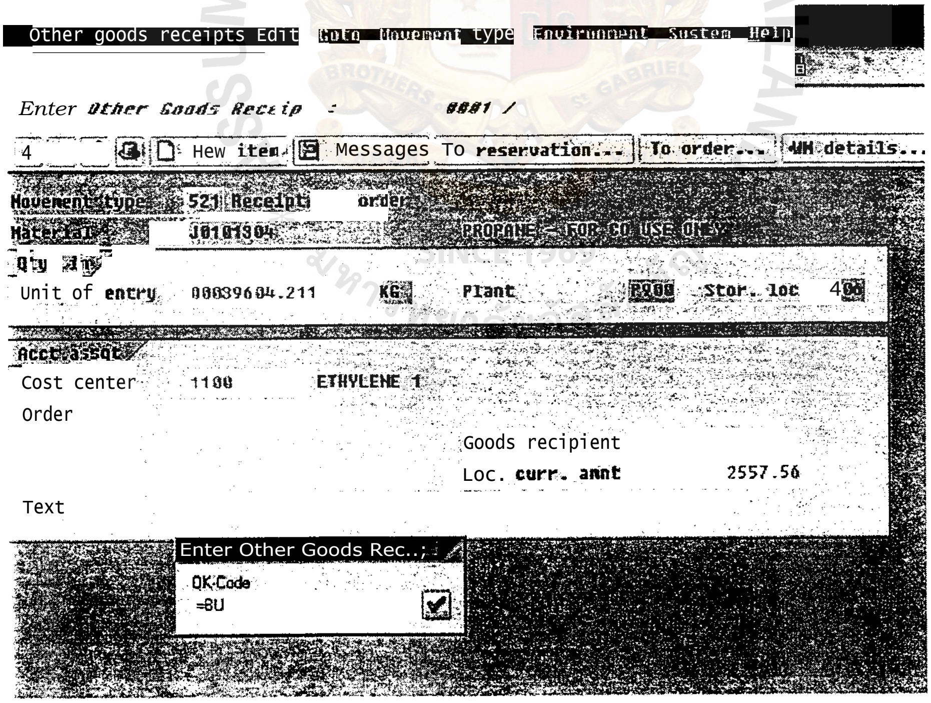
Cast iter 1100 ETHYLENE 1

Order

Item Material	Quantity	UnE	SLoc	Batch	Re Pint
A J0101304	39,604.211	"G	? **		
			P906		P900
			P906		P900
			P906		P900
			P906		P900
			P906		
			P906		P980
			P906		P900
			P906		P900
			P906		P900

Entry 1 of 1

=KPA



Enter Other Goods Receip : 0001 /

Movement type 521 Receipt order

Material J0101304 PROPANE - FOR CO USE ONLY

Qty

Unit of entry 00039604.211 KG Plant P900 Stor. loc 400

Cost center 1100 ETHYLENE 1

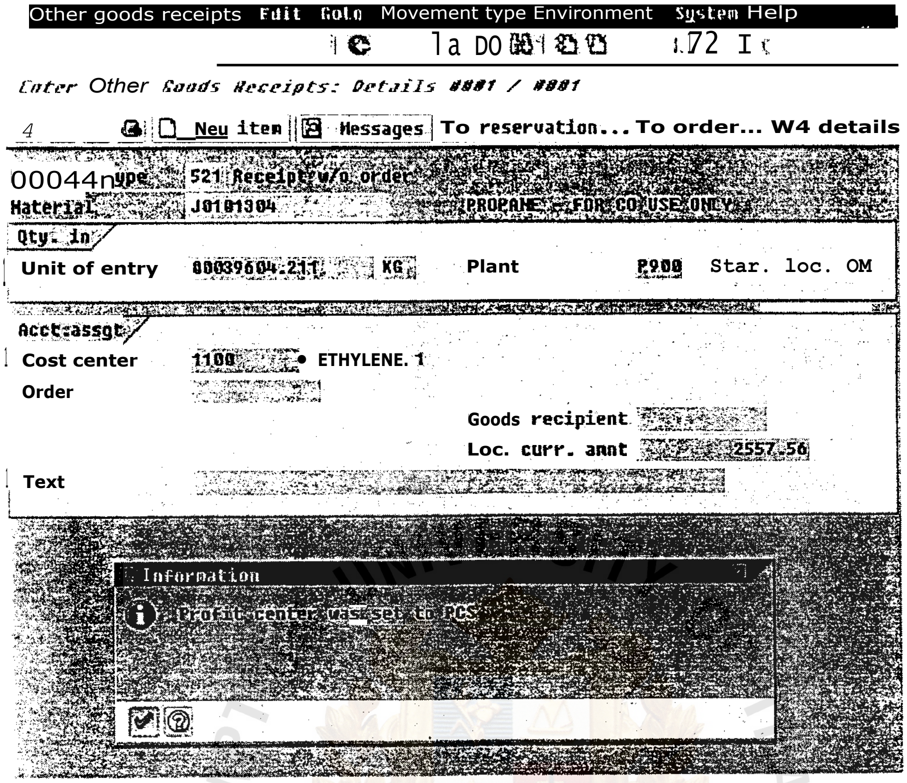
Order

Goods recipient

Loc. curr. amnt 2557.56

Text

=BU

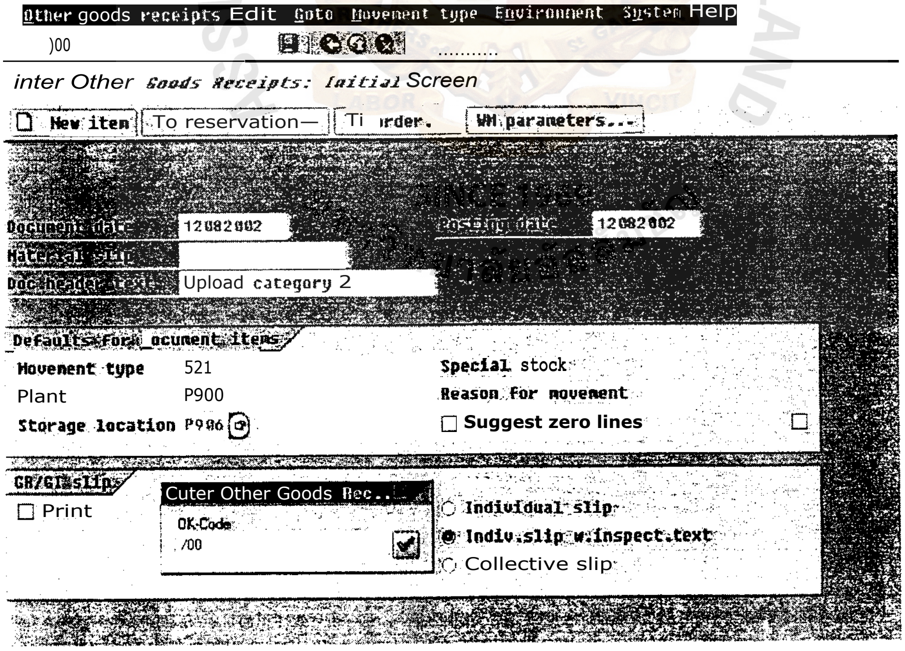

Other goods receipts Edit Goto Movement type Environment System Help
To reservation... To order... WM details
Neu item
Messages

Enter Other Goods Receipts: Details 0001 / 0001

Movement type	521 Receipt w/o order		
Material	J0101304	PROPANE - FOR CO USE ONLY	
Qty. in			
Unit of entry	00039604.211 KG	Plant	P900 Stor. loc. OM
Acct.assgt.			
Cost center	1100 ● ETHYLENE 1		
Order			
		Goods recipient	
		Loc. curr. amnt	2557.56
Text			

Information

Profit center was set to PCS


Other goods receipts Edit Goto Movement type Environment System Help
New item
To reservation...
To order...
WM parameters...

Enter Other Goods Receipts: Initial Screen

Document date	12082002	Posting date	12082002
Material slip			
Doc.header text	Upload category 2		
Defaults for document items			
Movement type	521	Special stock	
Plant	P900	Reason for movement	
Storage location	P906	□ Suggest zero lines	□

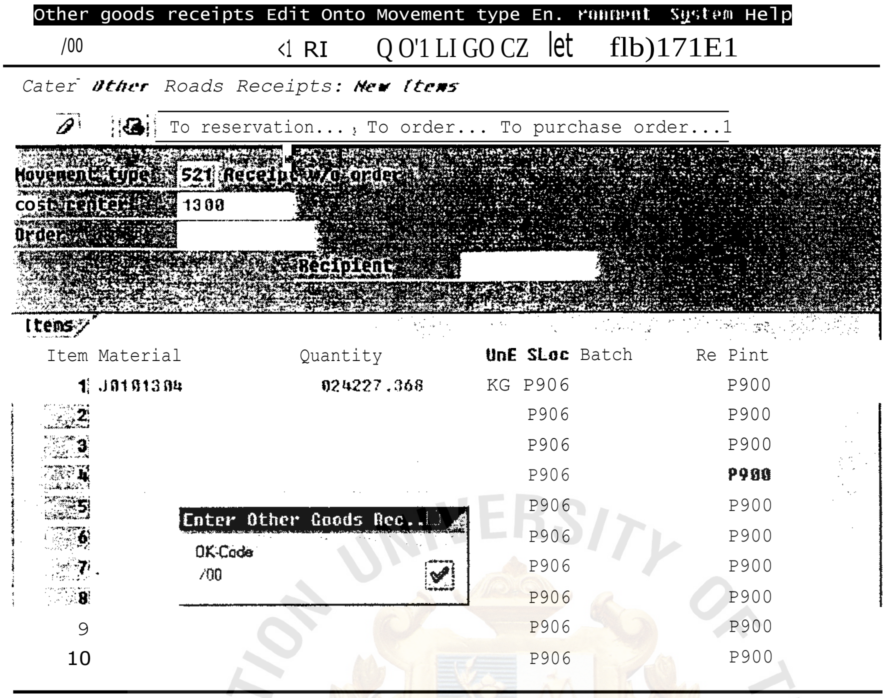
GR/GI slip

□ Print

Enter Other Goods Rec...

OK-Code
/00

○ Individual slip
● Indiv.slip w.inspect.text
○ Collective slip



Other goods receipts Edit Onto Movement type En. ronment System Help
/00
◁ RI Q O'1 LI GO CZ let flb)171E1

Cater Other Roads Receipts: New Items

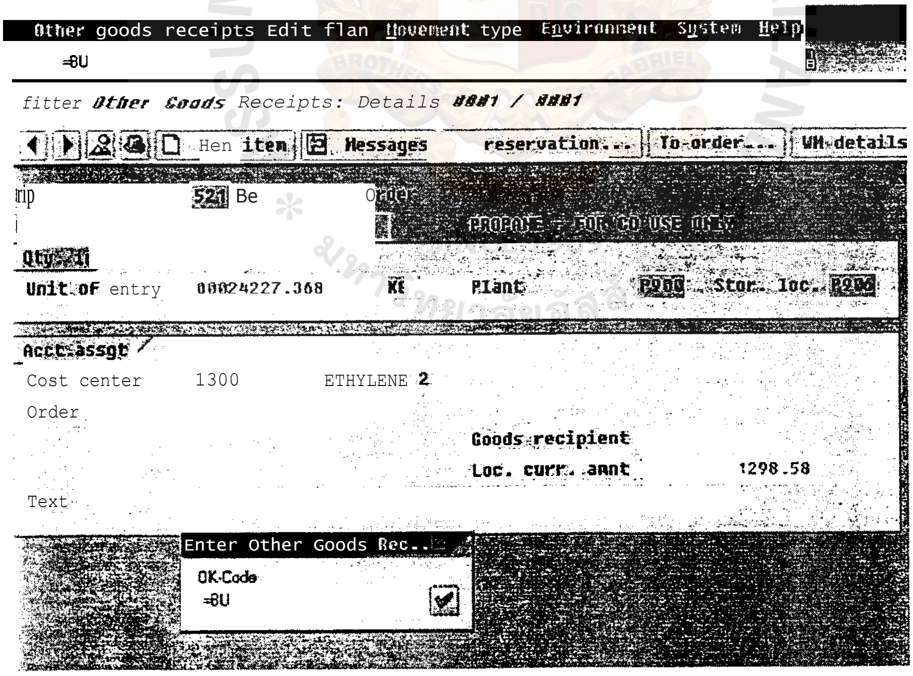
To reservation... ; To order... To purchase order...1

Movement type 521 Receipt w/o order
cost center 1300
Order
Recipient

Items

Item	Material	Quantity	UnE	SLoc	Batch	Re Pint
1	J0101304	024227.368	KG	P906		P900
2				P906		P900
3				P906		P900
4				P906		P900
5				P906		P900
6				P906		P900
7				P906		P900
8				P906		P900
9				P906		P900
10				P906		P900

Enter Other Goods Rec..
OK-Code
/00



Other goods receipts Edit flan Movement type Environment System Help
=BU

fitter Other Goods Receipts: Details 0001 / 0001

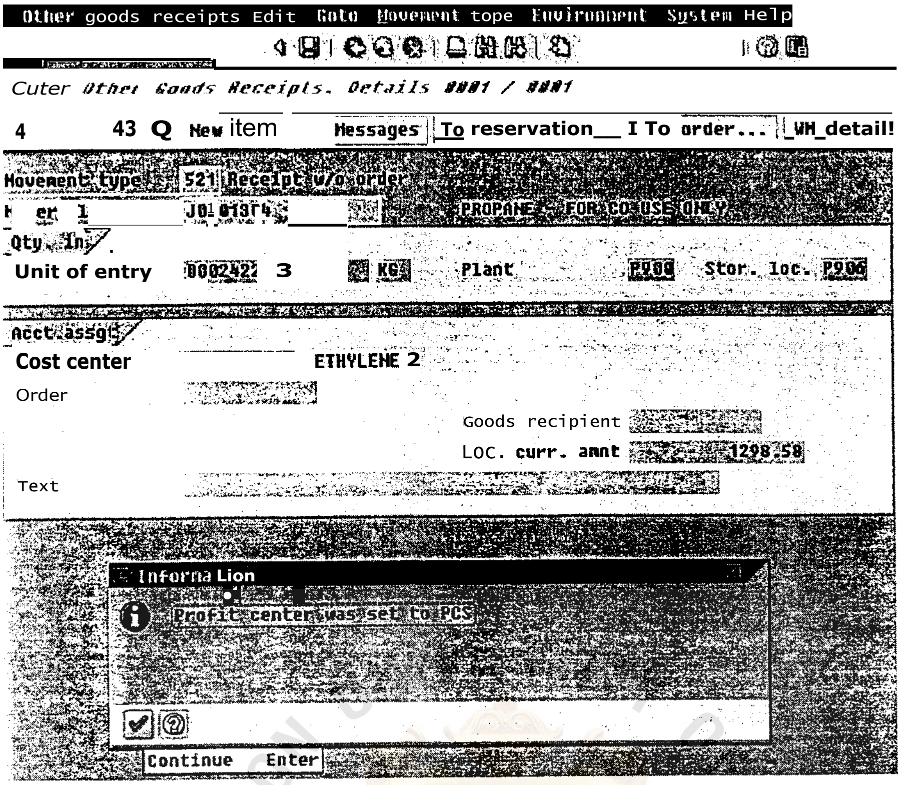
Hen iten Messages reservation... To-order... WH-details

rip 521 Be Order
PROPANE - FOR CO USE ONLY

Qty. 11
Unit of entry 00024227.368 KG Plant P900 Stor. loc. P906

Acct.assgt
Cost center 1300 ETHYLENE 2
Order
Goods recipient
Loc. curr. amnt 1298.58
Text

Enter Other Goods Rec..
OK-Code
=BU


Other goods receipts Edit Goto Movement tope Environment System Help

Cuter Other Goods Receipts. Details 0001 / 0001

4 43 Q New item
Messages
To reservation__ I To order...
_WH_detail!

Movement type 521 Receipt w/o order

H er 1 J0 013T4 PROPANE - FOR CO USE ONLY

Qty. In

Unit of entry 0002422 3 KG Plant P900 Stor. loc. P906

Acct.assgt

Cost center ETHYLENE 2

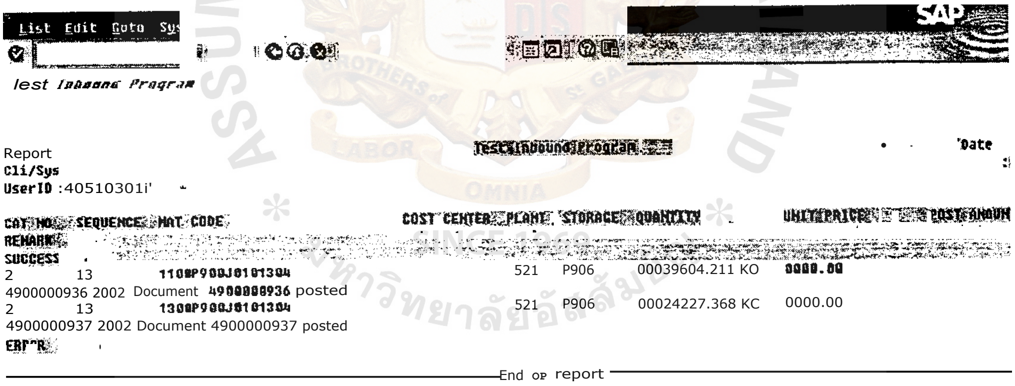
Order

Goods recipient

Loc. curr. amnt 1298.58

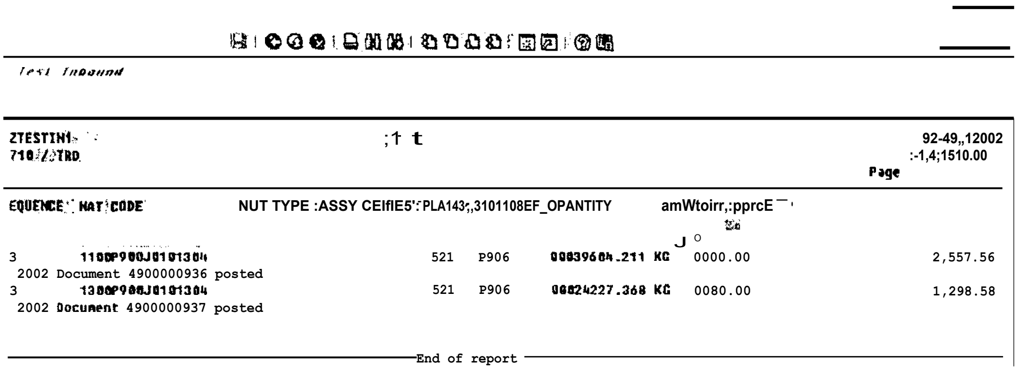
Text

Profit center was set to PCS



lest Inbound Program

```
Report                                   TestInboundProgram                        'Date
Cli/Sys
UserID :40510301i'

CAT NO  SEQUENCE  MAT CODE                COST CENTER  PLANT  STORAGE  QUANTITY          UNITPRICE        POSTAMOUN
REMARK
SUCCESS
2         13       1100P900J0101304              521     P906     00039604.211 KO   0000.00
4900000936 2002 Document 4900000936 posted
2         13       1300P900J0101304              521     P906     00024227.368 KC   0000.00
4900000937 2002 Document 4900000937 posted
ERR'R

                                         End oP report
```



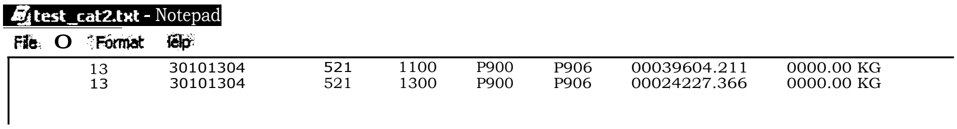
```
Test Inbound

ZTESTIN1                              ;t t                                              92-49,,12002
710/TRD                                                                                 :-1,4;1510.00
                                                                                   Page

EQUENCE  MAT CODE          NUT TYPE :ASSY CEIfIE5':PLA143;,3101108EF_OPANTITY      amWtoirr,:pprcE

3         1100P900J0101304                   521    P906     00039604.211 KG    0000.00          2,557.56
 2002 Document 4900000936 posted
3         1300P900J0101304                   521    P906     00024227.368 KG    0080.00          1,298.58
 2002 Document 4900000937 posted

--------------------------------------------End of report ------------------------------------------
```

- Example of incoming file for Run post program of category 2 : Finished Goods.

```
2        13       J0101304      521 1100 P900 P906 00039604.211
0000.00           KG
2        13       J0101304      521 1300 P900 P906 00024227.368
0000.00           KG
```

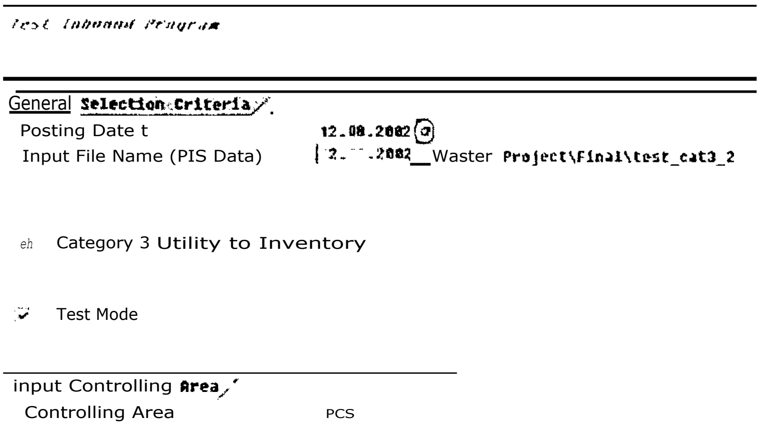
Format file :

2	: is category number 2
13	: is sequence of file
J0101304	: is Material number
521	: is Movement Type
1100,1300	: is Cost Center
P900	: is Storage Location
P906	: is Plant
39604.211	: is quantity
24227.268	: is quantity
KG	: is Unit of measurement



```
        13      30101304        521     1100    P900    P906    00039604.211    0000.00 KG
        13      30101304        521     1300    P900    P906    00024227.366    0000.00 KG
```

- Run test program of category 3 : Utility to Inventory, Program will generate report but no posting of data into SAP R/3 system.



Test Inbound Program

General Selection Criteria

Posting Date t 12.08.2002

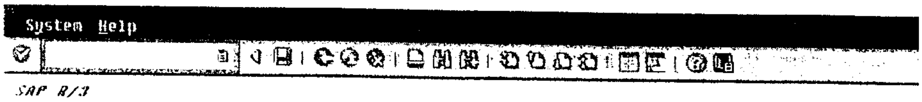
Input File Name (PIS Data) 2.   .2002__Waster Project\Final\test_cat3_2

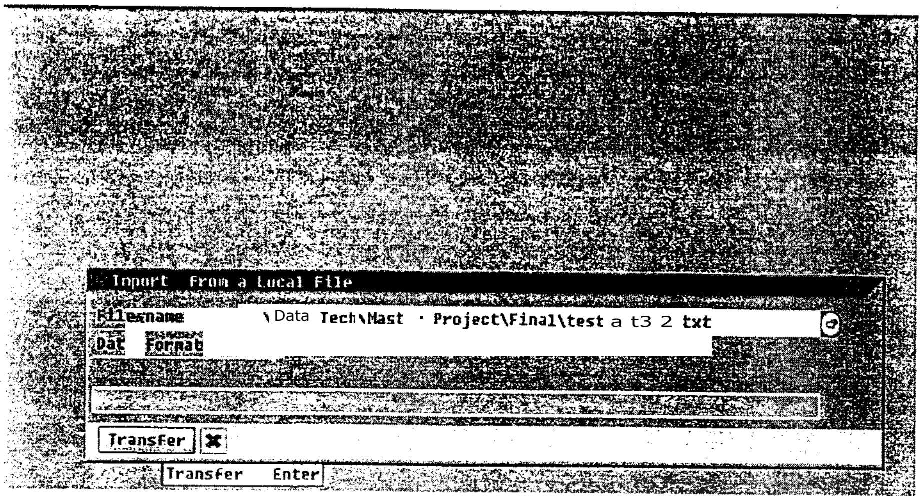
eh Category 3 Utility to Inventory

✓ Test Mode

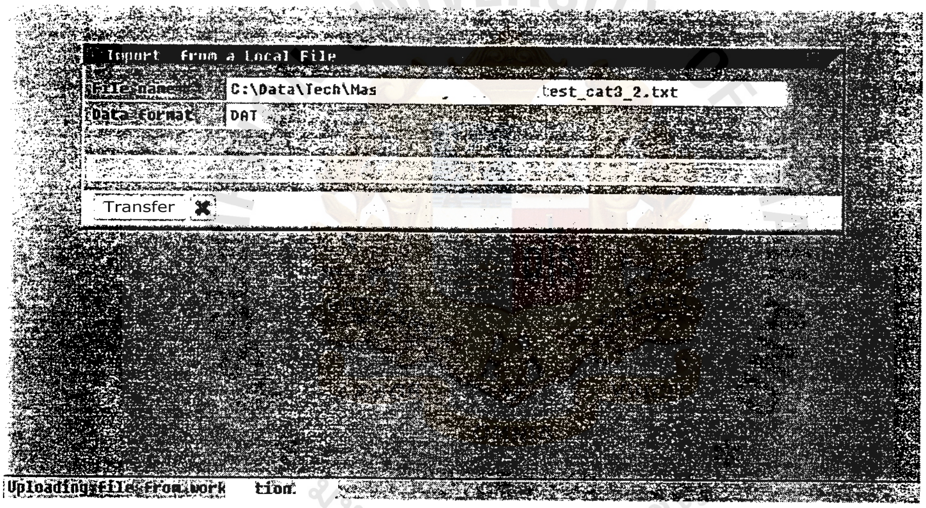
input Controlling Area

Controlling Area PCS





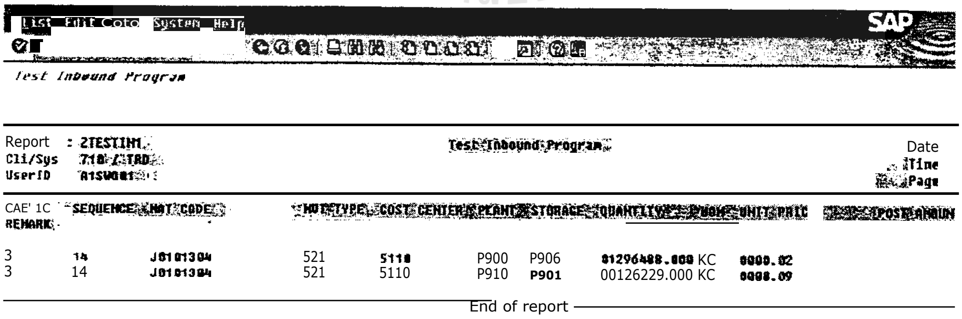
\Data Tech\Mast · Project\Final\test a t3 2 txt



C:\Data\Tech\Mas test_cat3_2.txt

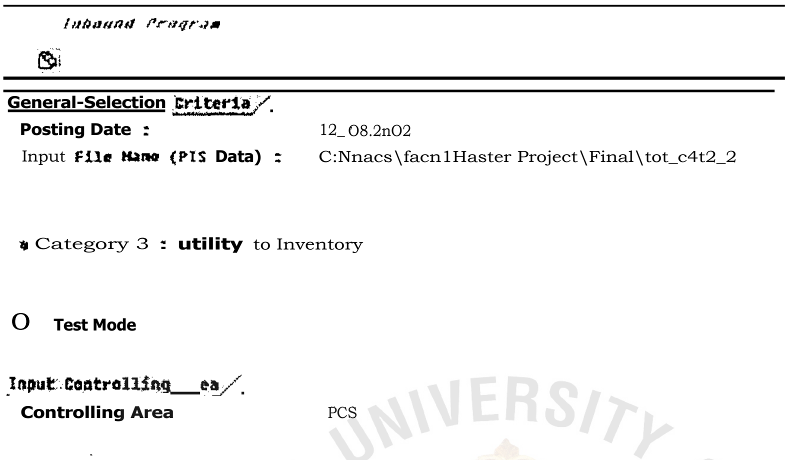
DAT

Uploading file from work tion.



```
Report   : ZTESTIM1                    Test Inbound Program                         Date
Cli/Sys    710 / TRD                                                                Time
UserID     A1SV001                                                                  Page

CAE' 1C   SEQUENCE MAT CODE       MOT TYPE COST CENTER PLANT STORAGE QUANTITY   UOM UNIT PRIC    POST AMOUN
REMARK

3         14      J0101304        521      5110        P900  P906    01296488.000 KC  0000.02
3         14      J0101304        521      5110        P910  P901    00126229.000 KC  0008.09
------------------------------------------- End of report ------------------------------------
```

- Run post program of category 3 : Utility to Inventory, Program will generate report and posting of data into SAP R/3 system. (Unmark Test Check box)



Inbound Program

General-Selection Criteria

Posting Date : 12_08.2n02

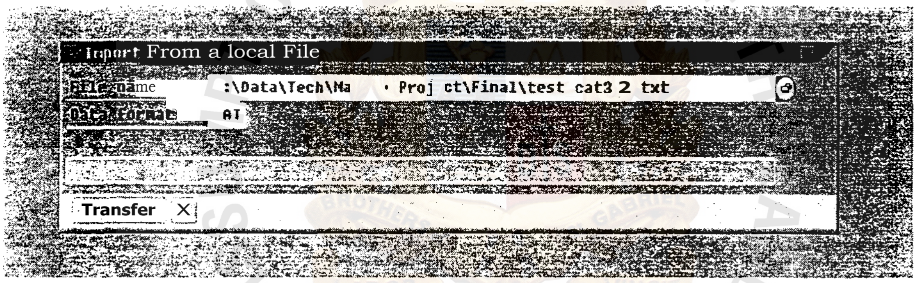
Input File Name (PIS Data) : C:Nnacs\facn1Haster Project\Final\tot_c4t2_2

● Category 3 : utility to Inventory

O Test Mode

Input Controlling ea

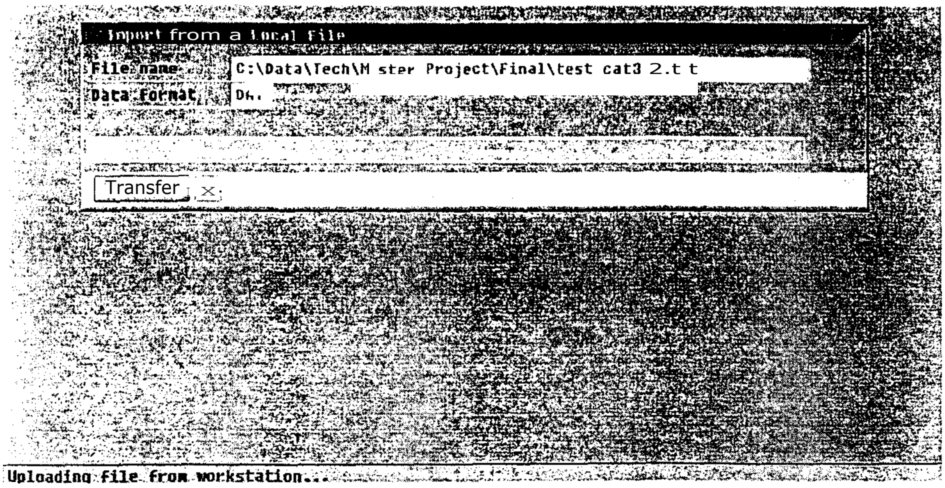
Controlling Area PCS



Import From a local File

File name :\Data\Tech\Ma · Proj ct\Final\test cat3 2 txt

Data Format AT

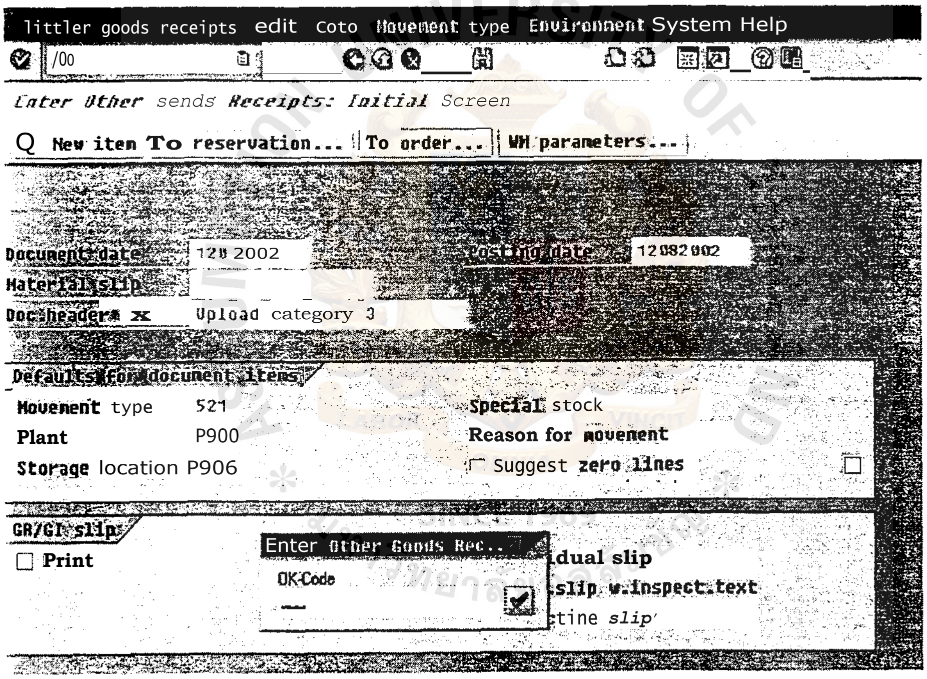


Import from a local file

File name: C:\Data\Tech\M ster Project\Final\test cat3 2.t t

Data format: DA.

Uploading file from workstation...



littler goods receipts edit Coto Movement type Environment System Help
/00

Enter Other sends Receipts: Initial Screen

New item To reservation... To order... WM parameters...

Document date: 128 2002

Posting date: 12082002

Material slip:

Doc.header x: Upload category 3

Defaults for document items

Movement type: 521

Special stock

Plant: P900

Reason for movement

Storage location: P906

□ Suggest zero lines

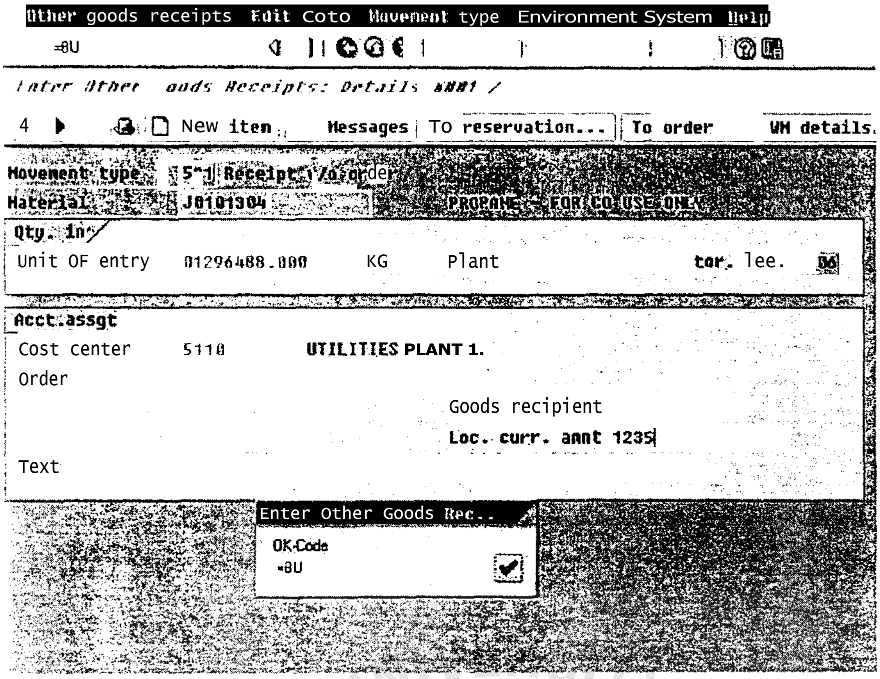
GR/GI slip

□ Print

Enter Other Goods Rec...

OK-Code

☑
dual slip
slip w.inspect.text
tine slip


Other goods receipts Edit Goto Movement type Environment System Help
=BU
New item Messages To reservation... To order WM details.

Enter Other Goods Receipts: Details 0001 /

Movement type 5~1 Receipt w/o order

Material J0101304 PROPANE - FOR CO USE ONLY

Qty. in

Unit OF entry 01296488.000 KG Plant Stor. loc. 06

Acct.assgt

Cost center 5110 UTILITIES PLANT 1.

Order

Goods recipient

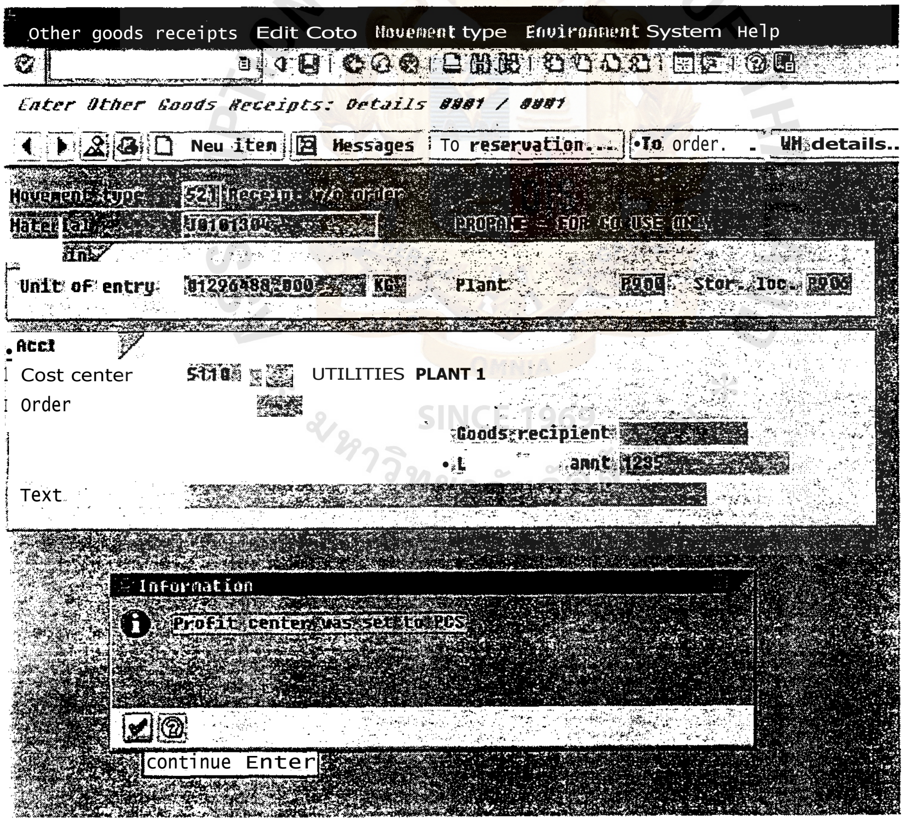
Loc. curr. amnt 1235

Text

Enter Other Goods Rec..

OK-Code

=BU


Other goods receipts Edit Goto Movement type Environment System Help
Neu item Messages To reservation... To order. WM details..

Enter Other Goods Receipts: Details 0001 / 0001

Movement type 521 Receipt w/o order

Material J0101304 PROPANE - FOR CO USE ONLY

Qty. in

Unit of entry 01296488.000 KG Plant P900 Stor. loc. P900

Acct

Cost center 5110 UTILITIES PLANT 1

Order

Goods recipient

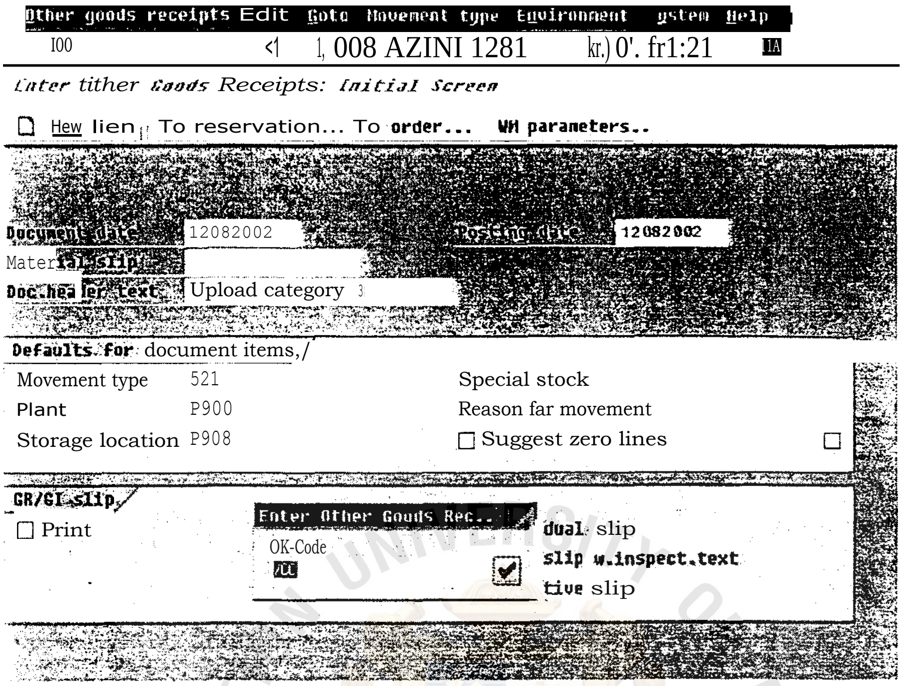
Loc. curr. amnt 1235

Text

Information

Profit center was set to PCS

continue Enter



Other goods receipts Edit Goto Movement type Environment System Help
I00
1, 008 AZINI 1281
kr.) 0'. fr1:21

Enter Other Goods Receipts: Initial Screen

New item To reservation... To order... WM parameters..

Document date 12082002
Posting date 12082002
Material slip
Doc.header text Upload category 3

Defaults for document items

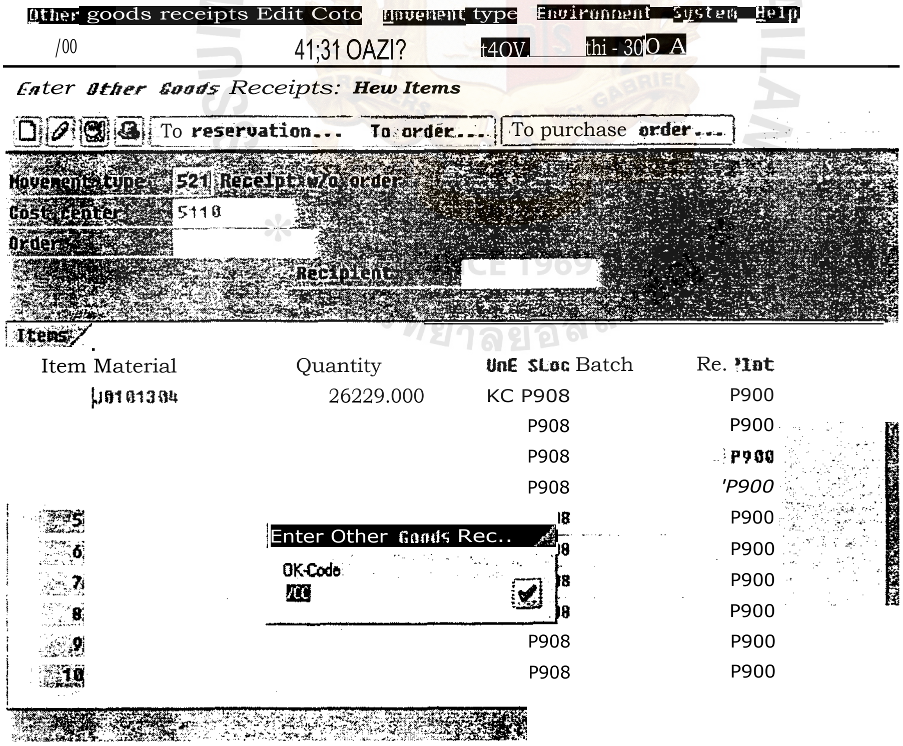
Movement type	521	Special stock	
Plant	P900	Reason far movement	
Storage location	P908	□ Suggest zero lines	□

GR/GI slip

□ Print

Enter Other Goods Rec..
OK-Code
/00
☑

dual slip
slip w.inspect.text
tive slip



Other goods receipts Edit Goto Movement type Environment System Help
/00
41;31 OAZI?

Enter Other Goods Receipts: New Items

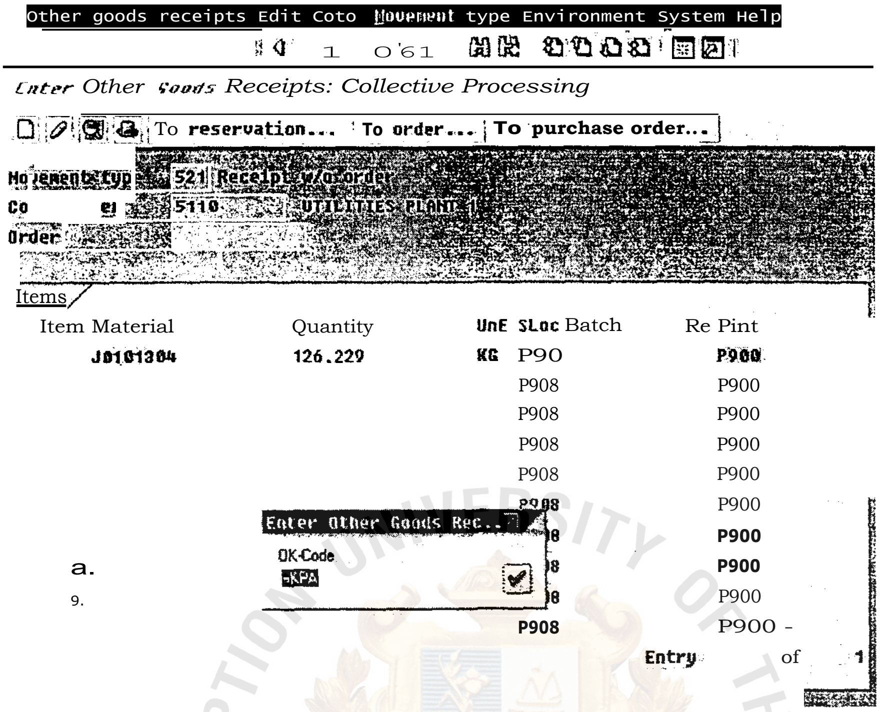
To reservation... To order... To purchase order...

Movement type 521 Receipt w/o order
Cost center 5110
Order
Recipient

Items

Item	Material	Quantity	UnE	SLoc	Batch	Re. Plnt
	J0101304	26229.000	KC	P908		P900
				P908		P900
				P908		P900
				P908		P900
5				8		P900
6				8		P900
7				8		P900
8				8		P900
9				P908		P900
10				P908		P900

Enter Other Goods Rec..
OK-Code
/00
☑


Other goods receipts Edit Coto Movement type Environment System Help
1 0'61
To reservation... To order... To purchase order...
Items
Enter Other Goods Rec..
OK-Code
-KPA

Enter Other Goods Receipts: Collective Processing

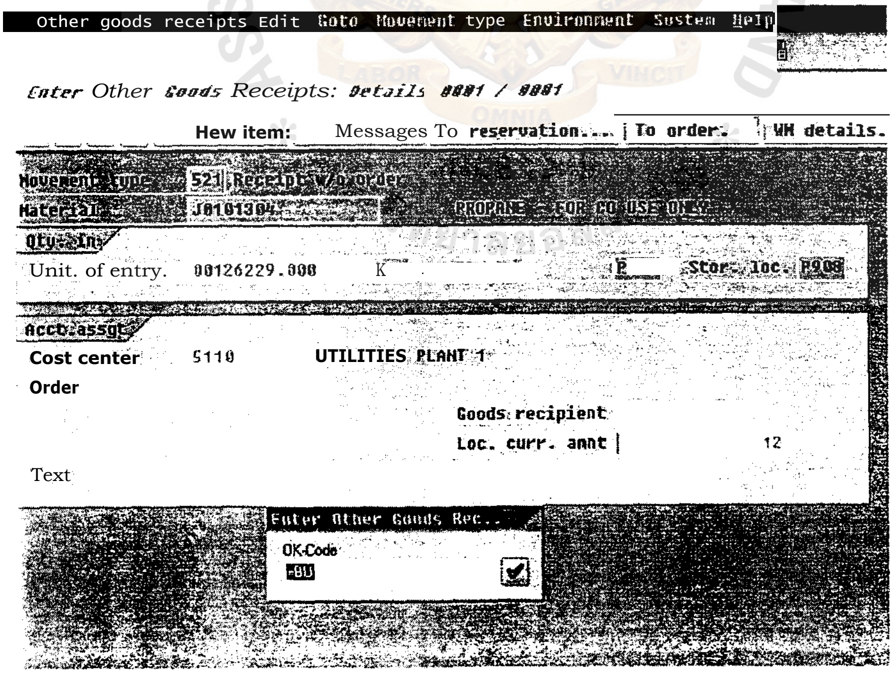
Movement typ 521 Receipt w/o order
Co el 5110 UTILITIES PLANT 1
Order

Item Material	Quantity	UnE	SLoc	Batch	Re Pint
J0101304	126.229	KG	P90		P900
			P908		P900
			P908		P900
			P908		P900
			P908		P900
			P908		P900
			8		P900
			8		P900
			8		P900
			P908		P900 -

Entry of 1

a.

9.


Other goods receipts Edit Goto Movement type Environment System Help
New item: Messages To reservation... To order. WM details.
Qty. in
Acct.assgt
Enter Other Goods Rec..
OK-Code
-BU

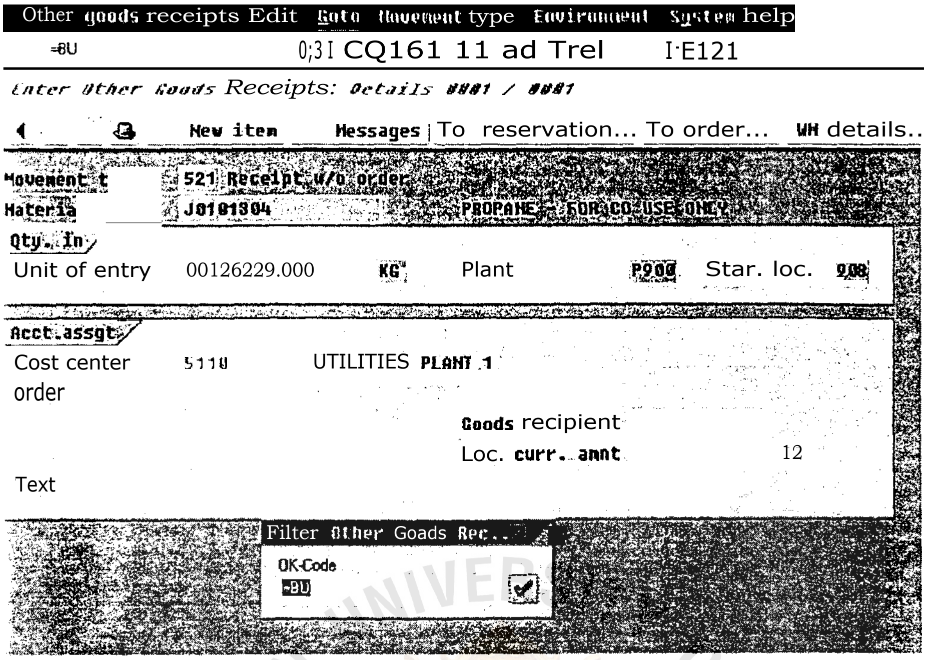
Enter Other Goods Receipts: Details 0001 / 0001

Movement type 521 Receipt w/o order
Material J0101304 PROPANE - FOR CO USE ONLY

Unit. of entry. 00126229.000 K P Stor. loc. P908

Cost center 5110 UTILITIES PLANT 1
Order
Goods recipient
Loc. curr. amnt 12

Text


Other goods receipts Edit Goto Movement type Environment System help
=BU
0;3I CQ161 11 ad Trel I·E121
New item
Messages
To reservation...
To order...
WH details..

Enter Other Goods Receipts: Details 0001 / 0001

Movement t 521 Receipt w/o order
Materia J0101304 PROPANE - FOR CO USE ONLY

Qty. in

Unit of entry 00126229.000 KG Plant P900 Star. loc. 908

Acct.assgt

Cost center 5110 UTILITIES PLANT 1

order

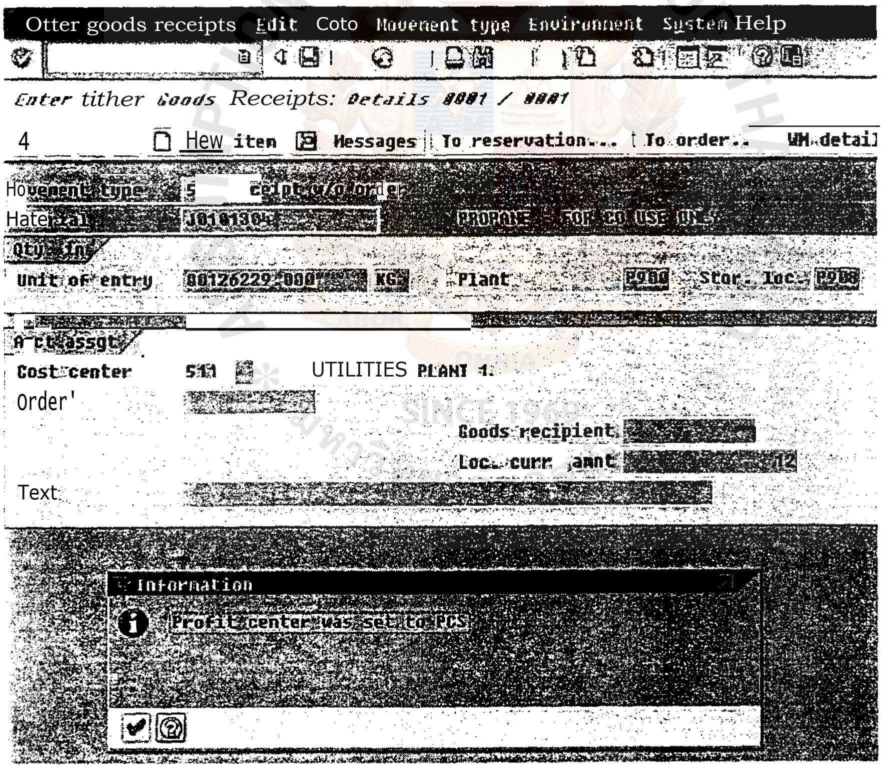
Goods recipient

Loc. curr. amnt 12

Text

Filter Other Goads Rec...

OK-Code
=BU


Otter goods receipts Edit Coto Movement type Environment System Help
4
New item
Messages
To reservation...
To order..
WH detail

Enter tither Goods Receipts: Details 0001 / 0001

Movement type 5 ceipt w/o order
Material J0101304 PROPANE - FOR CO USE ONLY

Qty. in

Unit of entry 00126229.000 KG Plant P900 Stor. loc. P908

Acct.assgt

Cost center 511 UTILITIES PLANT 1

Order'

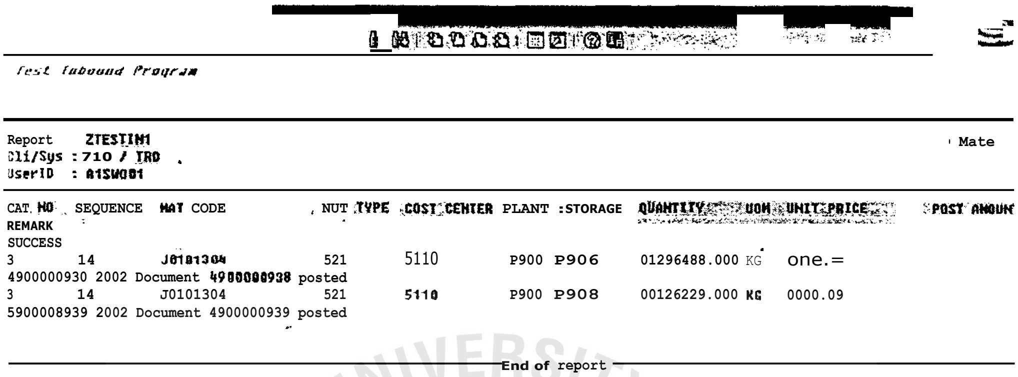
Goods recipient

Loc. curr. amnt 12

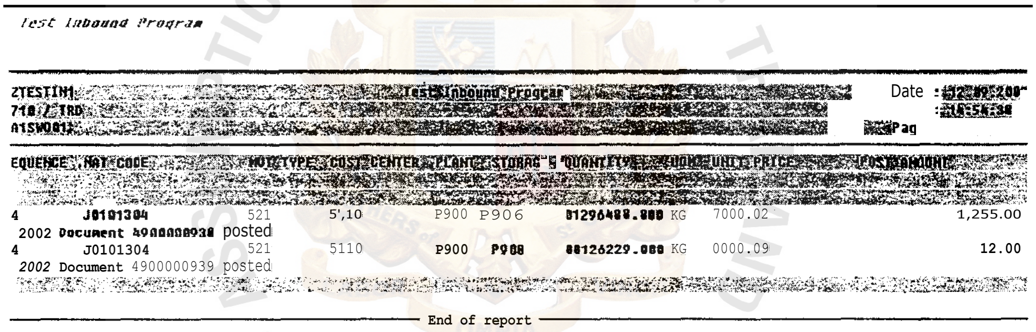
Text

Information

Profit center was set to PCS



```
Test Inbound Program

Report    ZTESTIN1                                                                                  Mate
Cli/Sys : 710 / TRD
UserID  : A1SW001

CAT. NO   SEQUENCE   MAT CODE              NUT TYPE  COST CENTER  PLANT  STORAGE  QUANTITY        UOM  UNIT PRICE     POST AMOUN
REMARK
SUCCESS
3         14         J0101304              521       5110         P900   P906     01296488.000    KG   one.=
4900000930 2002 Document 4900000938 posted
3         14         J0101304              521       5110         P900   P908     00126229.000    KG   0000.09
5900008939 2002 Document 4900000939 posted

------------------------------------------------ End of report ------------------------------------------------
```



```
Test Inbound Program

ZTESTIN1                              Test Inbound Program                                   Date :
710 / TRD                                                                                        :
A1SW001                                                                                      Pag

EQUENCE  MAT CODE          NUT TYPE  COST CENTER  PLANT  STORAG S QUANTITY      UOM  UNIT PRICE      POST AMOUNT

4        J0101304          521       5',10        P900   P906     01296488.000  KG   7000.02                1,255.00
 2002 Document 4900000938 posted
4        J0101304          521       5110         P900   P908     00126229.000  KG   0000.09                   12.00
 2002 Document 4900000939 posted

------------------------------------------ End of report ------------------------------------------
```

- Example of incoming file for Run post program of category 3 : Utility to Inventory,

3 14 J0101304 521 5110 P900 P906 01296488.000

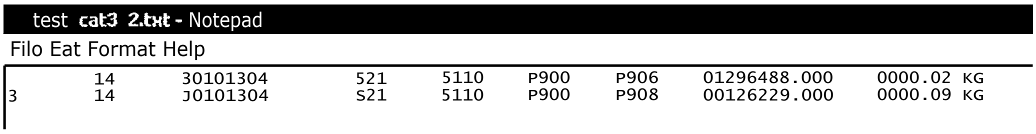
0000.02 KG

3 14 J0101304 521 5110 P900 P908 00126229.000

0000.09 KG

Format file :

3	: is category number 3
14	: is sequence of file
J0101304	: is Material number
521	: is Movement Type
5110	: is Cost Center
P900	: is Storage Location
P906,P908	: is Plant
1,296,488	: is quantity
126,229	: is quantity
KG	: is Unit of measurement

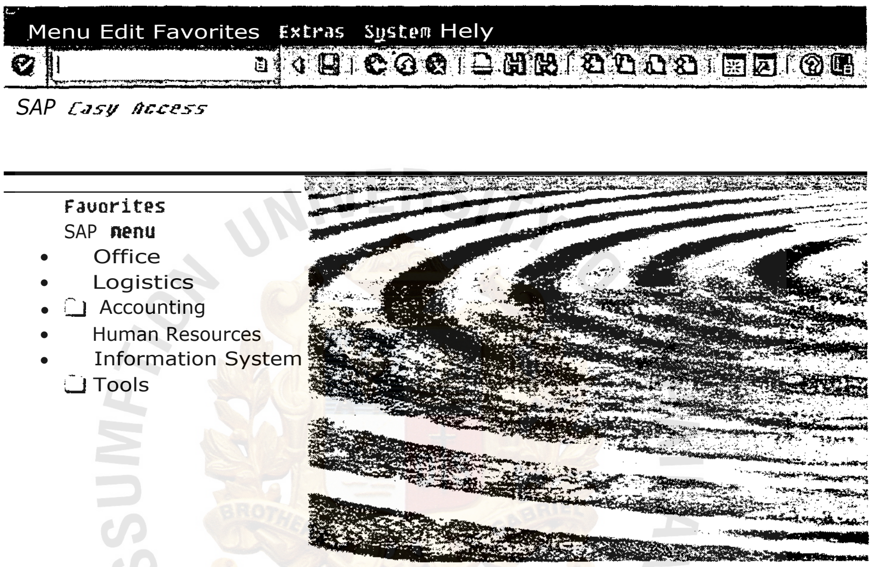


```
        14      30101304        521     5110    P900    P906    01296488.000    0000.02 KG
3       14      J0101304        S21     5110    P900    P908    00126229.000    0000.09 KG
```

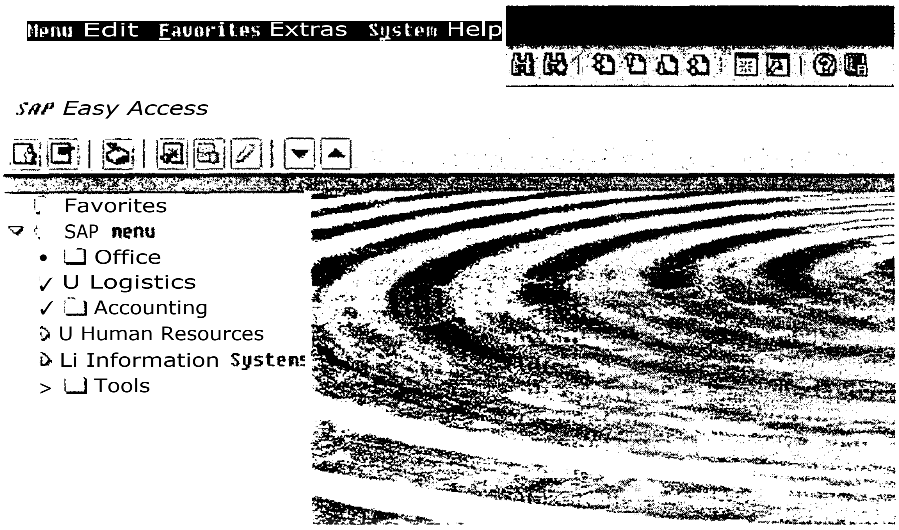
4.6.2 Screen Shot of Outbound Program:

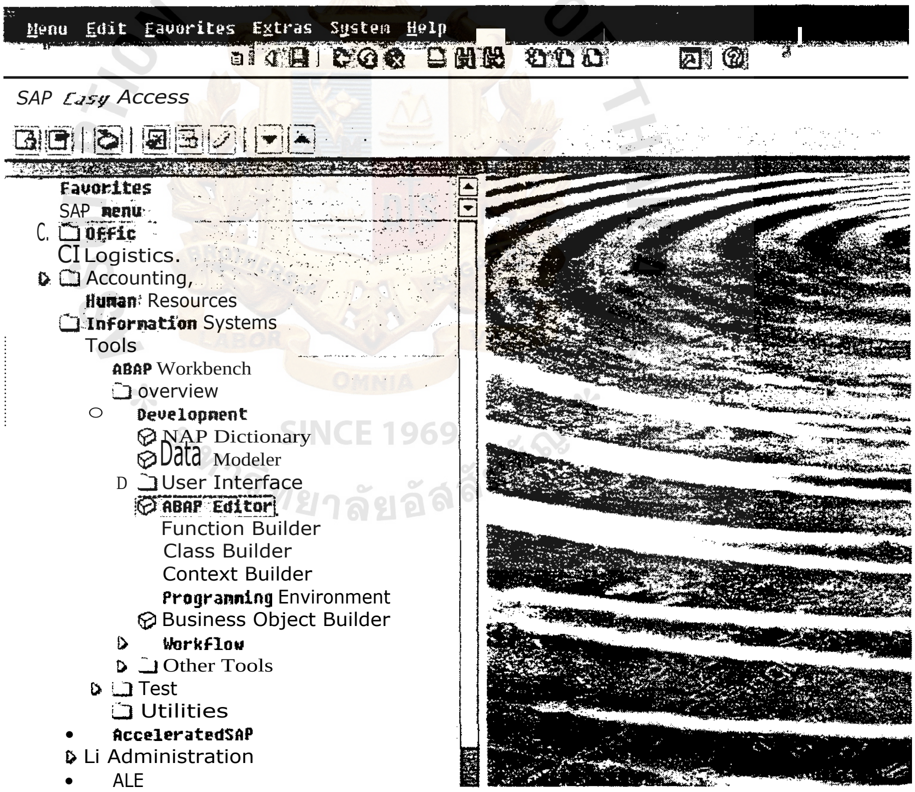
To run outbound program in SAP R13 system, user can select via transaction SE38 or run inbound program via SAP R/3 menu path. For inbound program steps can be explained as below in figure:

- Normal Screen of SAP R13 System.

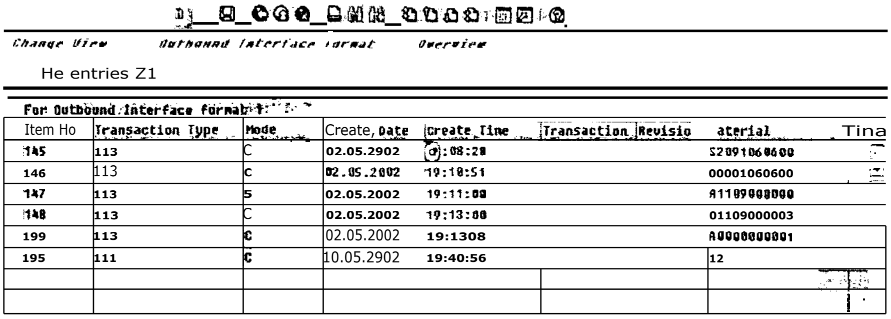


- Run program via transaction SE38 or via SAP R13 menu path.





- Interface database before run outbound program.

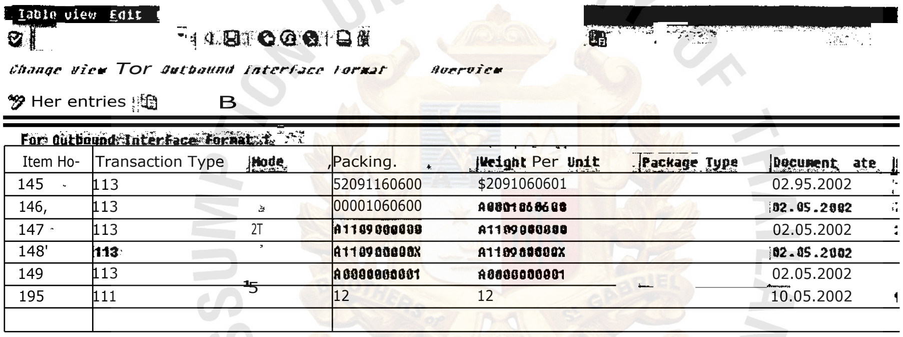


Change View Outbound Interface format Overview

He entries Z1

For Outbound Interface format 1

Item Ho	Transaction Type	Mode	Create, Date	Create Time	Transaction	Revisio	aterial	Tina
145	113	C	02.05.2902	[illegible]:08:28			S2091060600	
146	113	C	02.05.2002	19:10:51			00001060600	
147	113	5	02.05.2002	19:11:00			A11090000000	
148	113	C	02.05.2002	19:13:00			01109000003	
199	113	C	02.05.2002	19:1308			A00000000001	
195	111	C	10.05.2902	19:40:56			12	

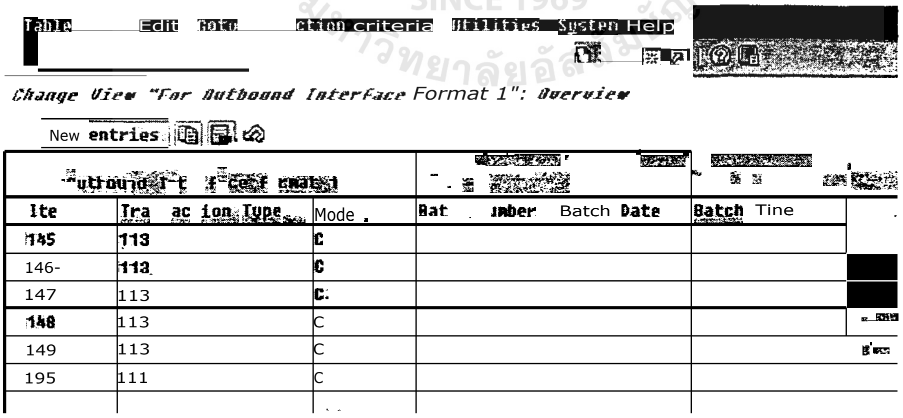


Change View For Outbound Interface Format Overview

Her entries B

For Outbound Interface Format 1

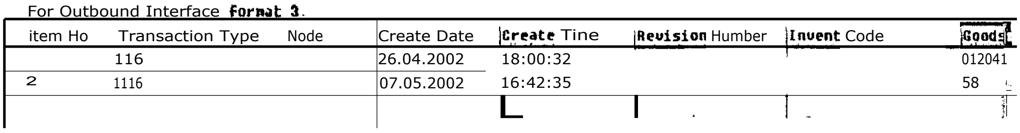
Item Ho-	Transaction Type	Mode	Packing.	Weight Per Unit	Package Type	Document ate
145	113		52091160600	$2091060601		02.95.2002
146,	113		00001060600	A00001060600		02.05.2002
147	113	2T	A11090000000	A11090000000		02.05.2002
148'	113		A1109000000X	A1109000000X		02.05.2002
149	113	5	A00000000001	A00000000001		02.05.2002
195	111		12	12		10.05.2002



Change View "For Outbound Interface Format 1": Overview

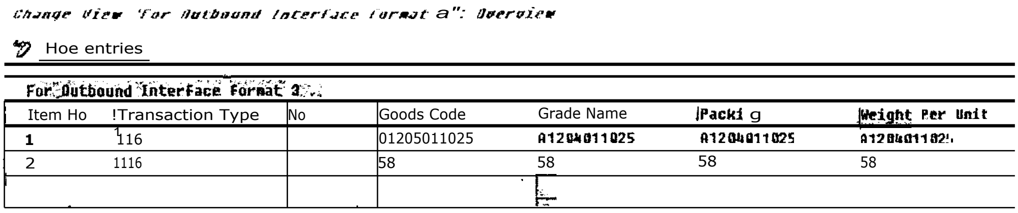
New entries

utroud I t f ceaf rmat 1				
Ite	Tra ac ion Type	Mode	Bat mber Batch Date	Batch Tine
145	113	C		
146-	113	C		
147	113	C:		
148	113	C		
149	113	C		
195	111	C		



For Outbound Interface Format 3.

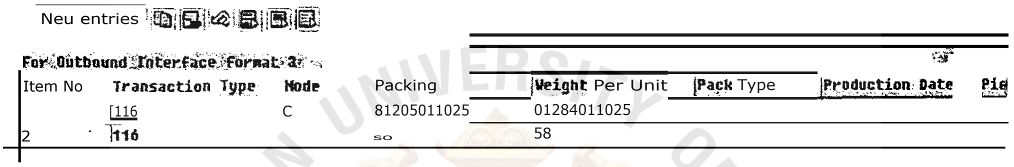
item Ho	Transaction Type	Node	Create Date	Create Tine	Revision Humber	Invent Code	Goods
	116		26.04.2002	18:00:32			012041
2	1116		07.05.2002	16:42:35			58



Change View "For Outbound Interface Format a": Overview

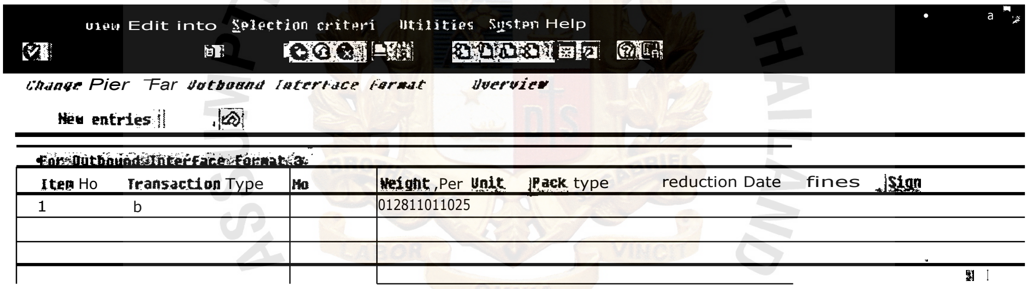
For Outbound Interface Format 3

Item Ho	!Transaction Type	No	Goods Code	Grade Name	Packi g	Weight Per Unit
1	116		01205011025	A1204011025	A1204011025	A12040118 2!
2	1116		58	58	58	58



For Outbound Interface Format 3

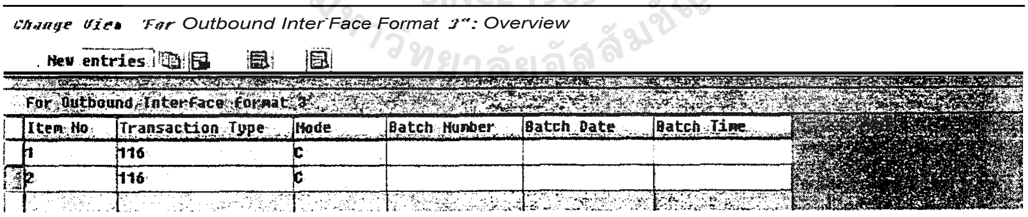
Item No	Transaction Type	Hode	Packing	Weight Per Unit	Pack Type	Production Date	Pie
	116	C	81205011025	01284011025			
2	116		so	58			



Change Pier "Far Outbound Interface Format Overview

For Outbound Interface Format 3

Item Ho	Transaction Type	Mo	Weight ,Per Unit	Pack type	reduction Date	fines	Sign
1	b		012811011025				

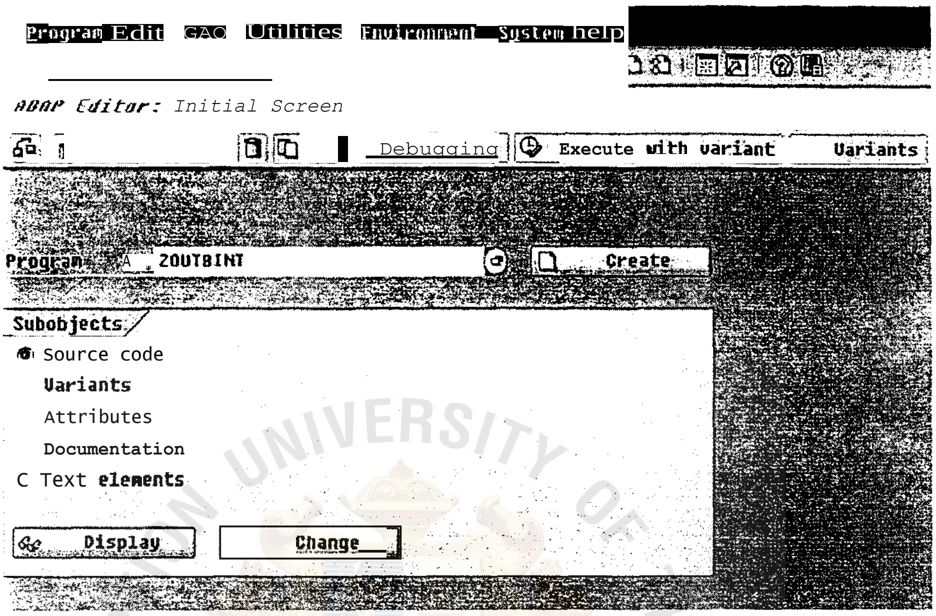


Change View "For Outbound Inter Face Format 3": Overview

For Outbound Interface Format 3

Item No	Transaction Type	Mode	Batch Number	Batch Date	Batch Time
1	116	C			
2	116	C			

- Input program name (ZOUTBINT) for Outbound program in ABAP editor and use function key (F8) to run program or click 🕒 to run program. (For example for Batch number 159)

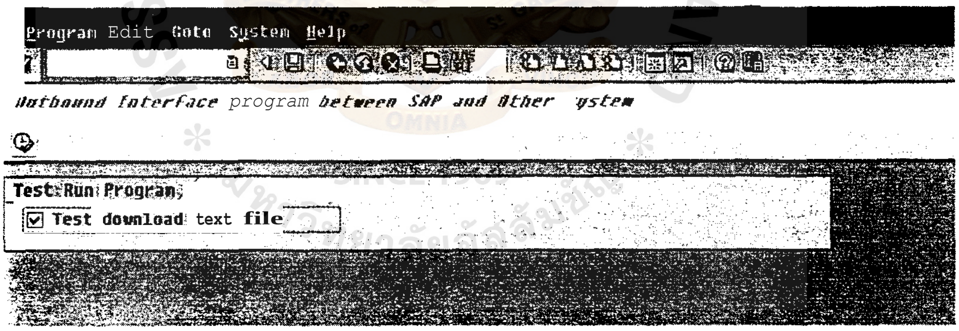


ABAP Editor: Initial Screen

Program ZOUTBINT

Subobjects

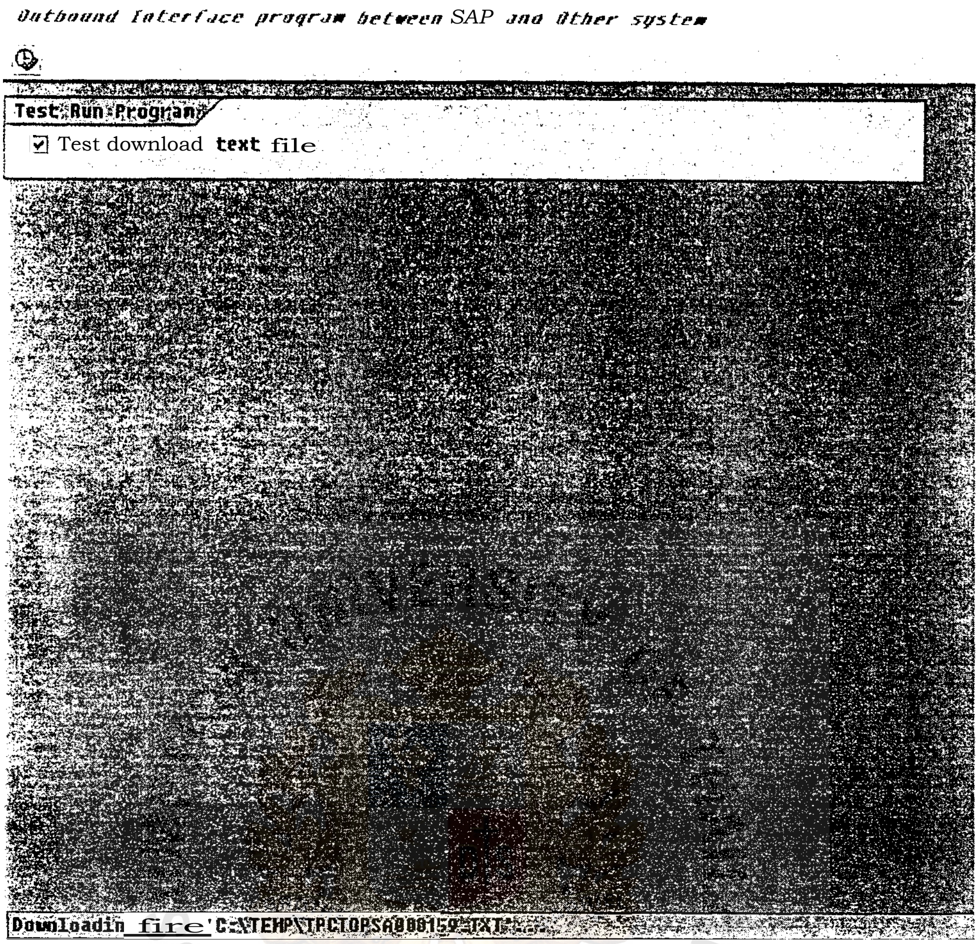
● Source code
Variants
Attributes
Documentation
C Text elements



Outbound Interface program between SAP and Other system

Test Run Program

☑ Test download text file

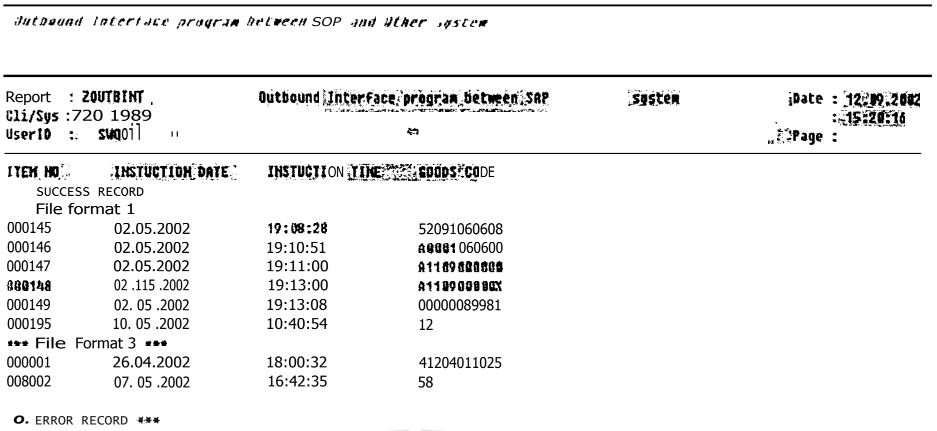


Outbound Interface program between SAP and Other system

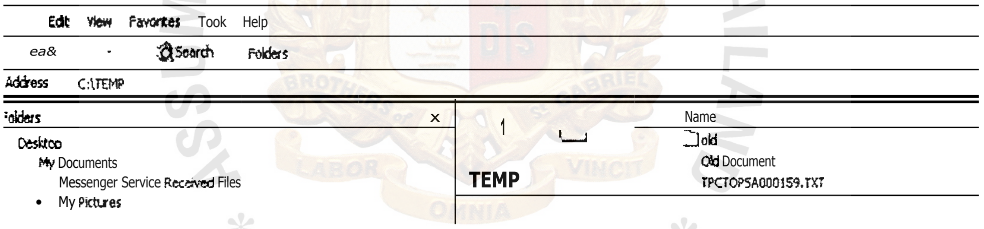
Test Run Program

☑ Test download text file

Downloadin_fire'C:\TEMP\TPCTOPSA000159.TXT'...



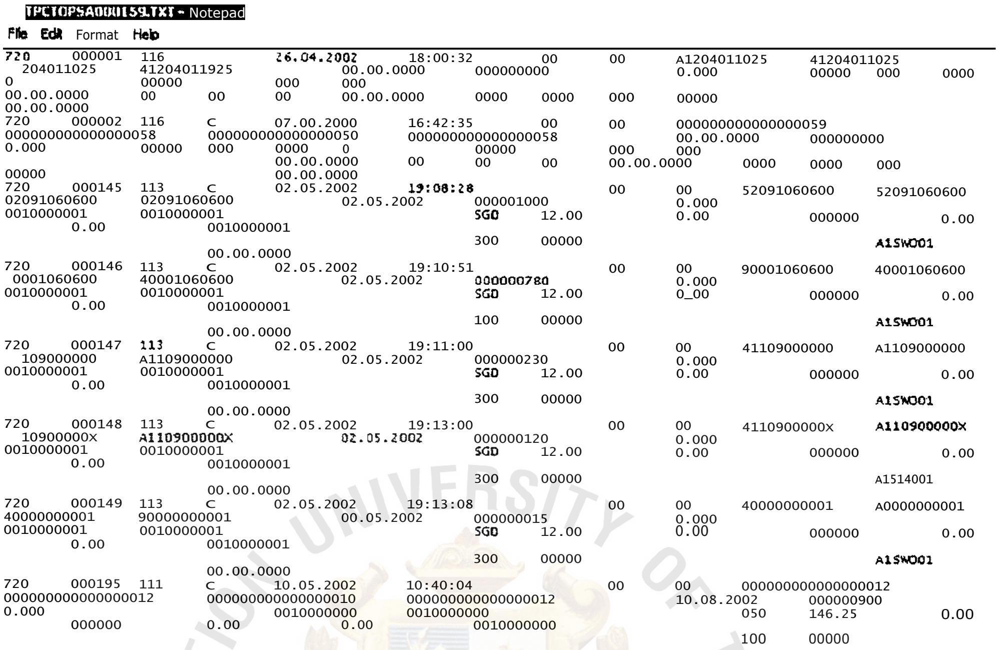
```
Outbound Interface program between SOP and Other system

Report   : ZOUTBINT .                Outbound Interface program between SAP        System          Date : 12.09.2002
Cli/Sys :720 1989                                                                                        : 15:20:16
UserID  :.  SWQ01l                                                                                  Page :

ITEM NO     INSTUCTION DATE     INSTUCTION TIME   GOODS CODE
   SUCCESS RECORD
   File format 1
000145        02.05.2002          19:08:28           52091060608
000146        02.05.2002          19:10:51           A0081060600
000147        02.05.2002          19:11:00           A1109000000
000148        02 .115 .2002       19:13:00           A1109000000X
000149        02. 05 .2002        19:13:08           00000089981
000195        10. 05 .2002        10:40:54           12
*** File  Format 3 ***
000001        26.04.2002          18:00:32           41204011025
008002        07. 05 .2002        16:42:35           58

 O. ERROR RECORD ***
```



Address C:\TEMP

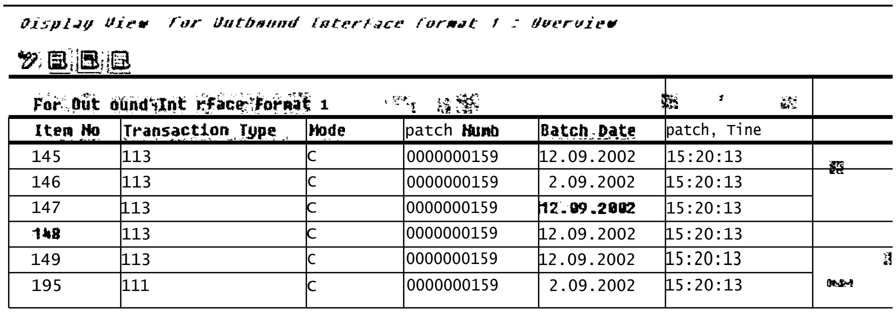
old
Old Document
TPCTOPSA000159.TXT

- Example of out going file after run outbound program.



```
720       000001    116                 26.04.2002          18:00:32            00        00        A1204011025         41204011025
 204011025          41204011925                   00.00.0000          000000000                     0.000               00000     000       0000
0                   00000               000       000
00.00.0000          00        00        00        00.00.0000          0000      0000      000       00000
00.00.0000
720       000002    116       C         07.00.2000          16:42:35            00        00        000000000000000059
000000000000000058            000000000000000050            000000000000000058                      00.00.0000          000000000
0.000               00000     000       0000      0                   00000               000       000
                                        00.00.0000          00        00        00        00.00.0000          0000      0000      000
00000
720       000145    113       C         02.05.2002          19:08:28                      00        00        52091060600         52091060600
02091060600         02091060600                   02.05.2002          000001000                     0.000
0010000001          0010000001                                        SGD       12.00               0.00                000000              0.00
          0.00                0010000001
                                                                      300       00000                                             A1SWO01
                              00.00.0000
720       000146    113       C         02.05.2002          19:10:51                      00        00        90001060600         40001060600
 0001060600         40001060600                   02.05.2002          000000780                     0.000
0010000001          0010000001                                        SGD       12.00               0_00                000000              0.00
          0.00                0010000001
                                                                      100       00000                                             A1SWO01
                              00.00.0000
720       000147    113       C         02.05.2002          19:11:00                      00        00        41109000000         A1109000000
  109000000         A1109000000                   02.05.2002          000000230                     0.000
0010000001          0010000001                                        SGD       12.00               0.00                000000              0.00
          0.00                0010000001
                                                                      300       00000                                             A1SWO01
                              00.00.0000
720       000148    113       C         02.05.2002          19:13:00                      00        00        4110900000X         A110900000X
  10900000X         A110900000X                   02.05.2002          000000120                     0.000
0010000001          0010000001                                        SGD       12.00               0.00                000000              0.00
          0.00                0010000001
                                                                      300       00000                                             A1514001
                              00.00.0000
720       000149    113       C         02.05.2002          19:13:08                      00        00        40000000001         A0000000001
40000000001         90000000001                   00.05.2002          000000015                     0.000
0010000001          0010000001                                        SGD       12.00               0.00                000000              0.00
          0.00                0010000001
                                                                      300       00000                                             A1SWO01
                              00.00.0000
720       000195    111       C         10.05.2002          10:40:04                      00        00        000000000000000012
000000000000000012            000000000000000010            000000000000000012                      10.08.2002          000000900
0.000                                   0010000000          0010000000                                        050       146.25              0.00
          000000              0.00                0.00                0010000000
                                                                                                              100       00000
```

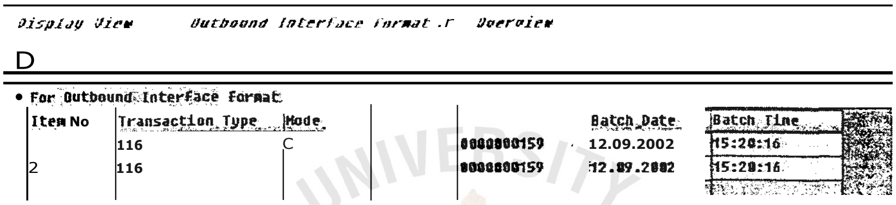
- Interface database after run outbound program.



Display View For Outbound Interface Format 1 : Overview

For Outbound Interface Format 1

Item No	Transaction Type	Mode	patch Numb	Batch Date	patch, Tine
145	113	C	0000000159	12.09.2002	15:20:13
146	113	C	0000000159	2.09.2002	15:20:13
147	113	C	0000000159	12.09.2002	15:20:13
148	113	C	0000000159	12.09.2002	15:20:13
149	113	C	0000000159	12.09.2002	15:20:13
195	111	C	0000000159	2.09.2002	15:20:13



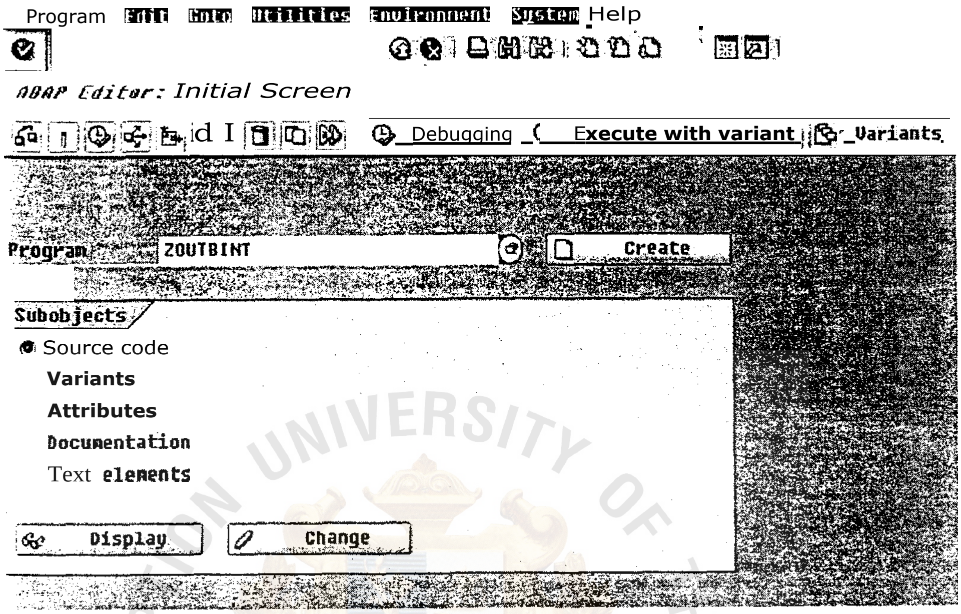
Display View Outbound Interface Format .r Overview

D

- For Outbound Interface Format

Item No	Transaction Type	Mode			Batch Date	Batch Time
	116	C		0000000159	12.09.2002	15:20:16
2	116			0000000159	12.09.2002	15:20:16

- Input program name (ZOUTBINT) for Outbound program in ABAP editor and use function key (F8) to run program or click — to run program. (For example for Batch number 160)

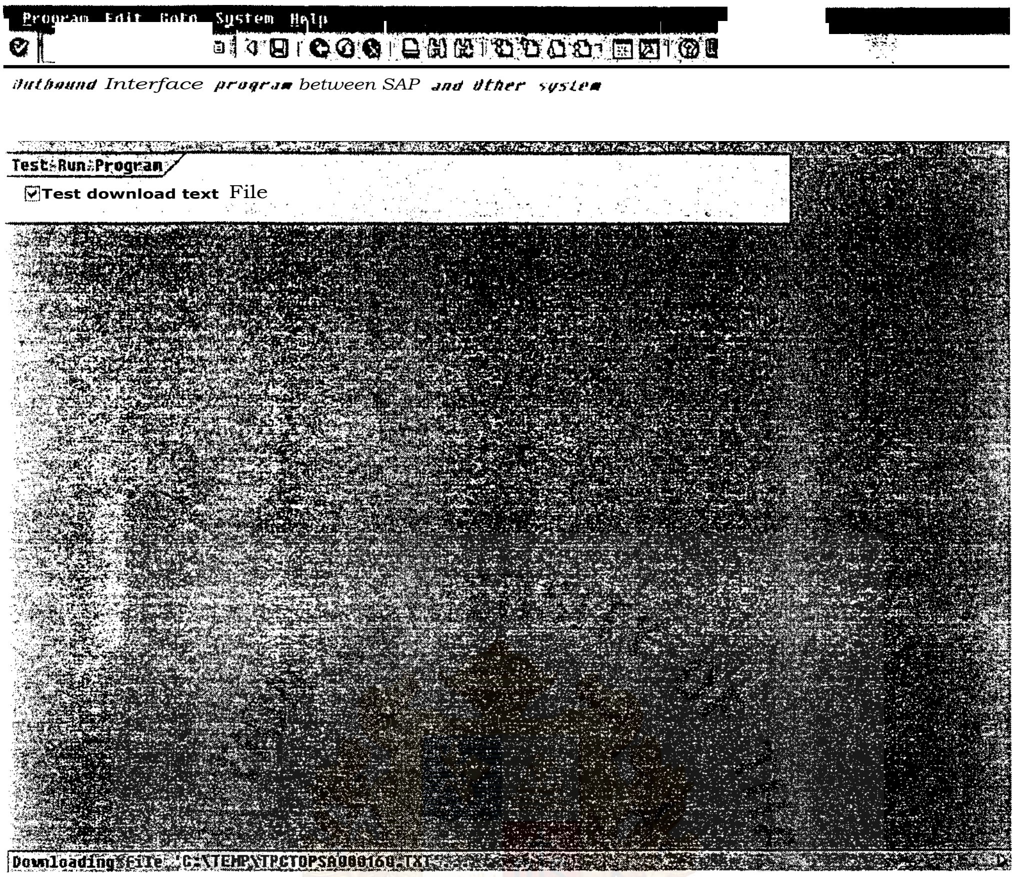


ABAP Editor: Initial Screen

Program ZOUTBINT

Subobjects

● Source code
○ Variants
○ Attributes
○ Documentation
○ Text elements

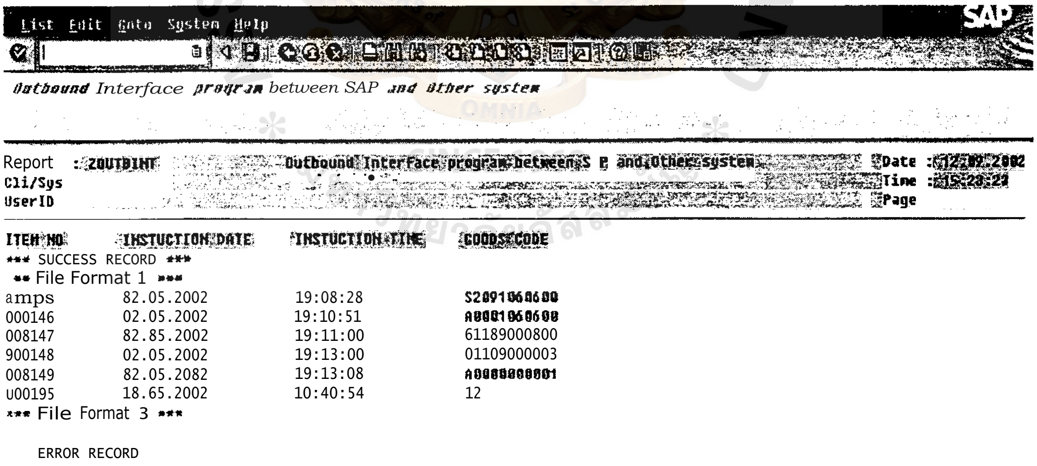


Outbound Interface program between SAP and Other system

Test Run Program

☑Test download text File

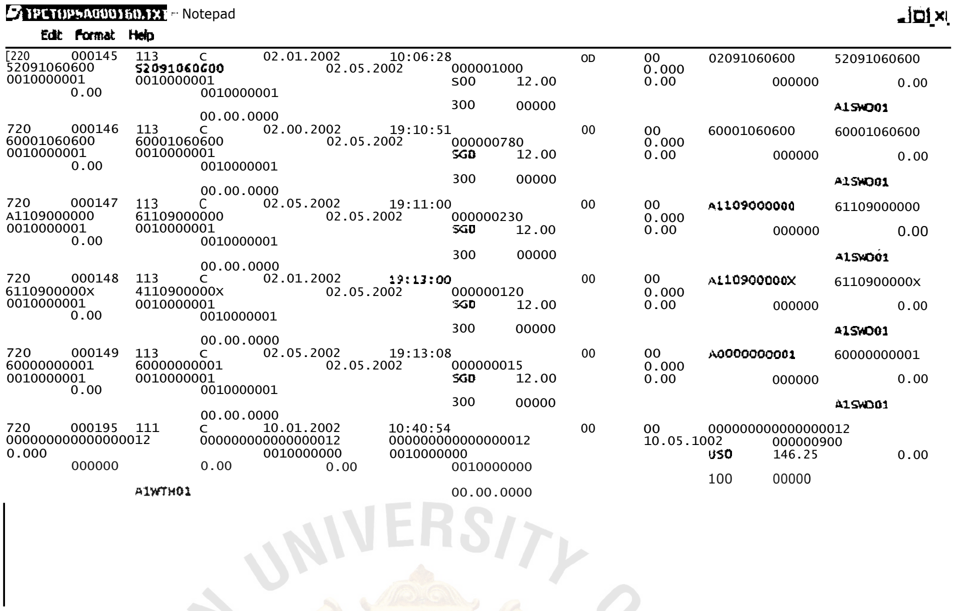
Downloading File 'C:\TEMP\TPCTOPSA000160.TXT'



Outbound Interface program between SAP and Other system

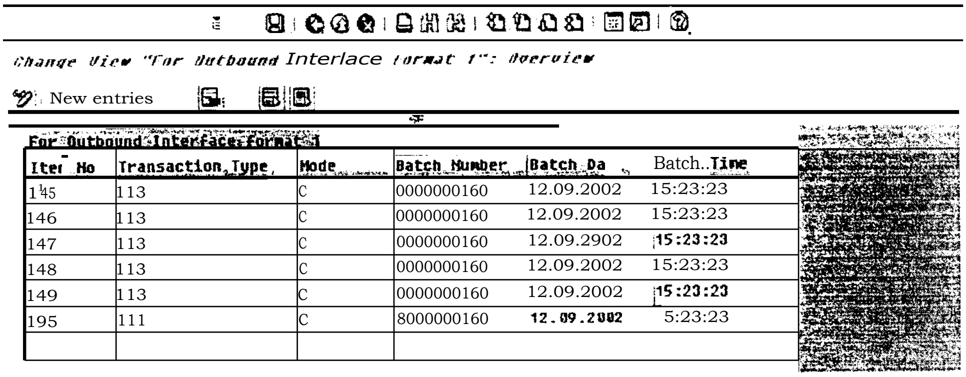
```
Report   : ZOUTBINT        Outbound Interface program between S P and Other system      Date :12.02.2002
Cli/Sys                                                                                  Time :15:28:22
UserID                                                                                   Page

ITEM NO       INSTUCTION DATE     INSTUCTION TIME     GOODS CODE
*** SUCCESS RECORD ***
 ** File Format 1 ***
amps          82.05.2002          19:08:28            S2091060600
000146        02.05.2002          19:10:51            A0001060600
008147        82.85.2002          19:11:00            61189000800
900148        02.05.2002          19:13:00            01109000003
008149        82.05.2082          19:13:08            A0000000001
U00195        18.65.2002          10:40:54            12
*** File Format 3 ***

    ERROR RECORD
```

- Example of out going file after run outbound program.



```
[220    000145  113     C       02.01.2002      10:06:28                OD      00      02091060600     52091060600
52091060600     S2091060600             02.05.2002      000001000               0.000
0010000001      0010000001                              S00     12.00           0.00            000000          0.00
        0.00            0010000001
                                                        300     00000                                   A1SWO01
                        00.00.0000
720     000146  113     C       02.00.2002      19:10:51                00      00      60001060600     60001060600
60001060600     60001060600             02.05.2002      000000780               0.000
0010000001      0010000001                              SGD     12.00           0.00            000000          0.00
        0.00            0010000001
                                                        300     00000                                   A1SWO01
                        00.00.0000
720     000147  113     C       02.05.2002      19:11:00                00      00      A1109000000     61109000000
A1109000000     61109000000             02.05.2002      000000230               0.000
0010000001      0010000001                              SGD     12.00           0.00            000000          0.00
        0.00            0010000001
                                                        300     00000                                   A1SWO01
                        00.00.0000
720     000148  113     C       02.01.2002      19:13:00                00      00      A110900000X     6110900000X
6110900000X     4110900000X             02.05.2002      000000120               0.000
0010000001      0010000001                              SGD     12.00           0.00            000000          0.00
        0.00            0010000001
                                                        300     00000                                   A1SWO01
                        00.00.0000
720     000149  113     C       02.05.2002      19:13:08                00      00      A0000000001     60000000001
60000000001     60000000001             02.05.2002      000000015               0.000
0010000001      0010000001                              SGD     12.00           0.00            000000          0.00
        0.00            0010000001
                                                        300     00000                                   A1SWO01
                        00.00.0000
720     000195  111     C       10.01.2002      10:40:54                00      00      000000000000000012
000000000000000012      000000000000000012      000000000000000012              10.05.1002      000000900
0.000                           0010000000      0010000000                              USO     146.25          0.00
        000000          0.00            0.00            0010000000
                                                                                        100     00000
                A1WTH01                                 00.00.0000
```

- Interface database after running outbound program.



Change View "For Outbound Interlace Format 1": Overview

New entries

For Outbound Interface Format 1

Iter No	Transaction Type	Mode	Batch Number	Batch Da	Batch..Time
145	113	C	0000000160	12.09.2002	15:23:23
146	113	C	0000000160	12.09.2002	15:23:23
147	113	C	0000000160	12.09.2902	15:23:23
148	113	C	0000000160	12.09.2002	15:23:23
149	113	C	0000000160	12.09.2002	15:23:23
195	111	C	8000000160	12.09.2002	5:23:23

4.7 Cost Analysis

For the cost implementation of EAT is very expensive, "Average costs to tie together only two major applications can run in the $1.5M to $4M range and have reached $10M in some instances" –AberdeenGroup. For the information from eAI Journal in implementation EAI application around $6,525,245 based on CrossWorld Application as in below table:

	EAI
First Project	
Architecture Costs	
Software licenses	$700,000
Hardware	$90,000
Architectureal implementation	$840,000
Total	$1,630,000
Interface **DevelopmentTime**	
Analysis and design time	8.78 days
Detailed design, build and test time	13.50 days
System test time	7.25 days
Total Time	29.53 days
Application development rate	$1,000/day
Cost per interface	$29,525
Nominal interface costs	$5,639,275
Saving from reuse of interface	$744,030
Total	$4,895,245
Total Costs- First Project	$6,525,245

Source : eAI Journal

Table 4-1 Integration cost of EAI base on CrossWorld application

The average days to implement Application Interface program between 2 systems is 45 days for each system including design and so testing program based on complexity is difficult.

This project provid customized application integration at a low cost. That can be compared with the US dollar and Thai baht because of this basic model can be re-used for the source code.

For the cost analysis based on US dollars, with the average development rate $1,000 per day, so total amount of this customized application integration is around $45,000 within 45 days. For the new application that needs to integrate needs only to write data into text file format that matches with SAP R/3 system. So the effort estimates around 45 days same as developed in SAP R/3 system.

So total costs of integration for both side is $90,000. This prototype customized application integration excludes the license of the new application that needs to integrate with SAP R/3 system.

Company	Baht/day
PriceWaterhouseCooper	29,000
Accenture	25,000
IT One	15,000
Contacter at Siam Cement	8,000
Contacter at Shell via Accenture	3,000

Table 4-2 Development Rate in Thai Market

For the cost analysis based on Thai Baht, with the average development rate for developing ABAP, which is the editor of SAP R/3, in Thai market is 15,000 Baht per day, so total amount of this customized application integration for both side is 1,350,000 Baht.

So, for both Dollar and Thai Baht, Customized Application Integration is cheaper than EAI application in market with the same requirements and the number of days for development that is 45 days.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This project provides the basic model of customized application integration to integrate SAP R/3 system with other systems, which SAP R/3 as a key driver for this integration. This model can serve a company that already has SAP R/3 system in integration with new application, or planning to implement SAP R/3 system with their Legacy system.

This project provides the source code of inbound and outbound programs of SAP R/3 system, which is the major architecture of the application integration between SAP R/3 with other systems. This customized application integration used file transfer methodology in transferring data between systems, that makes the cost low, easy for changing code and minor change for existing functions. These inbound and outbound programs can support both of UNIX path connection and workstation or PC between SAP R/3 system and other systems.

This project simulated getting data from incoming file and post good issues and good receipt in SAP R/3 system, this project provided automatic posting in SAP R/3 system. SAP R/3 system will generate good movements documents in the SAP R/3 database. For the outbound program will select all data from interface database and send out going file to external system by using batch control number to be the key indicator.

This project provides the source code of inbound and outbound customized integration application program that can be used to integrate between SAP R/3 with new system.

By using File Transfer methodology can lower cost of implementation for this customized application integration. But it has disadvantage as using file transfer in real-time information, this project also can serve this near real-time information by using customized application integration that is automatic posting and SAP R/3 has function to support for run schedule program. But it will cause another problem, which is performance of the system will be slow. But in real life, some of the data can be run as batch job depending on company requirements.

5.2 Benefit

This project will benefit the company, which wants to implement application integration based on SAP R/3 with other systems in several ways:

1. This study will help company in providing the basic model of integration with SAP R/3 system.
2. This study provides the solution of customized application integration of SAP R/3 system at low cost.
3. This study provides the easy way in implementation by using file transfer methodology.
4. This study provides the minimized change for the existing system base on SAP R/3 system.

5. This study provides the connection of integration for both UNIX connection and to send and receive data file with other systems. This will be the choice of integration for the company.
6. This study will help company to identify the advantage and disadvantage of each Integration Methodology.
7. This study will help company in terms of reducing time-to market for implementing integration with the basic model of application integration.
8. This study will help company to enhance customer and supplier services in integrated new application.
9. This study will help company to transfer large amount of data between systems with support for batch job.
10. This study provides the re-use source code of main interface in SAP R/3 system with other systems.
11. This study will help company increase efficiency and company product/service.
12. This study will help company to enhance competitive advantage to implement the suitable application for company.
13. This study provides the customized application integration customized to the prototype company requirements.

5.3 Limitation

The limitations of this project are:

1. Real-Time information because this project uses transfer file methodology that is cause for non real-time information but it is nearly real-time which is supported by SAP R/3's function but if it need to use SAP R/3's function, it might cause another issue that is performance issue for SAP R/3 because program needs to load the system all the time to trigger incoming file from other systems.
2. Number of connections between systems because this project needs to have one to one (1-1) connection for each system that includes 2 programs which are inbound and outbound program. So if the company has more than one system, company need to have another connection for the new connection.
3. Maintenance for this customized application integration will be hard because of the number of interface programs.

5.4 Further Research and Recommendation

Even though this project provides several benefits to company that wants to implement application integration, but this project still has many limitations in this customized application integration. So it is necessary for the future research to serve company requirements, which this project cannot support.

Further research should study about the architecture of each database system, which includes Oracle, Ariba, i2 and so on. This might help in design for the standard text file format in integration between systems and to reduce time to market of integration. And also we should study about ABAP application that is the Editor for SAP R/3 in coding automatic program with the real-time connection with no issue of system performance, and also study about how to trigger file from UNIX and how to trigger program SAP R/3 from UNIX with no system performance concerns. The number of connections should be the one concern for the further research study about how to reduce the number of integration between systems because it will be easy for maintenance period.

BIBLIOGRAPHY

1. Butler Group, Application Integration, 1999
2. Cherry Tree & Co., Extended Enterprise Applications, 2000
3. Aberdeen Group Inc., e-Business Infrastructure Integration: Practical Approaches, 2001
4. SeeBeyond., SeeBeyond Application, 1999
5. ACTel., Case Study: ACTel, 2000
6. Giga Information Group, Internet Application Integration : A New Market Emerges to Support Cross E-Business, 1999
7. SAP, MySAP Mobile Business, 2001
8. eAI Journal, Mark E. Atkins, Closing the EM GAP with data Integration, 2002
9. IDC Bulletin , Businessware Market Makers: Past Present and Future, 1998

APPENDIX A: Source Code of Inbound Customized Application Integration Program.

```
REPORT ZTESTIN1 LINE-SIZE 135
                LINE-COUNT 65
                NO STANDARD PAGE HEADING.

**********************************************************************
•  Program            : ZTESTIN1
•  Author             : Supaporn Wongwithit
•  Created            : 14/05/2002
*
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
•  Description        : Interface program between SAP and other system.
                    This program will transfer data from other system
                    into SAP system. This program will involve three
                    main categories.
*
                         1. In the first category : the interface
                    program will read data from input file,
                    calculate the amount to company code currency
                    and post these data in SAP.
*
                         2. In the second category : the program will
                    calculate and post price differences between
                    input and output quantities of the specified cost
                    center.
*
                         3. The last category : data from file will
                    be used for calculating and posting amount of
                    utility to inventory.
*
   Input               : Input file form work station in tab delimited
                         format. And input field as below :
*
   Output              : Display Success and Error records.
*
   Called from         : Transaction SE38 for run program.
*
   Called to           : N/A
*
   Includes            : N/A
*
   Function Modules    : N/A
*
   Logical Database    : N/A
*
   High Level Design   : N/A
*
*

**********************************************************************
       TABLES
*
**********************************************************************
TABLES :
  MARA  ,                                  " General Material Data
  TKA01 ,                                " Controlling Areas
  TKA03 ,                                " Statistical key figures
  CSKA  .                                " Cost Elements table

**********************************************************************
       DATA
*
**********************************************************************
  - - - -   Constants  - - - - - - - - - - - - - - - - - - - - - - - -
CONSTANTS :
  C_DEFAULT_PATH(50) TYPE C VALUE 'C:\'            " Path
  C_FILETYPE(3)      TYPE C VALUE 'DAT'            "  Filetype = DAT
  C UHP(8)           TYPE P DECIMALS 4
                            VALUE '1.0000'         " Ultra High
  C_HP(8)            TYPE P DECIMALS 4
                            VALUE '0.9274'         " High Pressure
  C MP(8)            TYPE P DECIMALS 4
                            VALUE '0.8684'       " Medium Pressure
  C LP(8)            TYPE P DECIMALS 4
                            VALUE '0.7896'            Low Pressure
  C_I(1)             TYPE C VALUE 'I'                 I
  C_EQ(2)            TYPE C VALUE 'EQ'                EQ=Equal
  C_KS(2)            TYPE C VALUE 'KS'                KS
  C_FE(2)            TYPE C VALUE 'FE'                Query command
  C_UPDATE(1)        TYPE C VALUE 'S'                 Update type
  C_00(2)            TYPE C VALUE '00'                00
  C_000(3)           TYPE C VALUE '000'               000
  C_04(2)            TYPE C VALUE '04'              " 04
  C_E(1)             TYPE C VALUE 'E'              English language
  C_A(1)             TYPE C VALUE 'A'              Foreground BDC
  C_X(1)             TYPE C VALUE 'X'                 X = MARK
  C_MASK(80)         TYPE C VALUE ',*.*,*.*           Get file
  C_MB1A(4)          TYPE C VALUE 'MB1A'              Good issues
  C_MB1C(4)          TYPE C VALUE 'MB1C'              Good receipt
  C_S(1)             TYPE C VALUE 'S'                 S = SUCCESS
  C_USD              LIKE TCURR-FCURR VALUE 'USD',    USD Currency
  C_SGD              LIKE TCURR-TCURR VALUE 'SGD',    SGD Currency
  C_060(3)           TYPE C VALUE '060'               060
  C_521(3)           TYPE C VALUE '521'               521
  C_901(3)           TYPE C VALUE '901'             " 901
  C_M7(2)            TYPE C VALUE 'M7'                M7
  C_COLON(1)         TYPE C VALUE ':'                 Colon
  C_TEST_HEAD(9)     TYPE C VALUE 'TEST MODE'         Test Mode
  C_SUCCESS_HEAD(7)  TYPE C VALUE 'SUCCESS'           Success
  C_ERROR_HEAD(5)    TYPE C VALUE 'ERROR'             Error
  C_ZERO(6)          TYPE C VALUE '000000'            000000
  C_EXGTYP           LIKE TCURR-KURST VALUE 'M'       Exchange TYPE
  C_PIS_TEXT         LIKE MKPF-BKTXT
                            VALUE 'Upload category
                                                    " Header text
  C_CHAR1(2)         TYPE C VALUE ' 1'              " Category 1
  C_CHAR2(2)         TYPE C VALUE ' 2'              " Category 2
  C_CHAR3(2)         TYPE C VALUE ' 3'              " Category 3
  C_C(1)             TYPE C VALUE 'C'               " CO product

  - - - - - Working areas  - - - - - - - - - - - - - - - - - - - - - - -

  - - - -   Internal table
TYPES : BEGIN OF TYP_TMPDATA,
          CATG(1)                TYPE C ,            " Category number
          SEQUE(2)                TYPE C  ,             Sequence number
          MATNR(18)               TYPE C  ,             Material Code
          BWART(3)                TYPE C  ,             Movement Type
          KOSTL(4)                TYPE C  ,             Cost center
          WERKS(4)                TYPE C                Plant
          LGORT(4)                TYPE C  ,             Storage Location
          ERFMG(12)               TYPE C  ,             Quantity
          PRICE(11)               TYPE C  ,             Unit Price (USD)
          UOM(3)                  TYPE C  ,             Unit of measure
        END OF TYP_TMPDATA.

TYPES : BEGIN OF TYP_TMPDATA1,
          CATG(1)                 TYPE C  ,             Category number
          SEQUE(2)                TYPE C  ,             Sequence number
          KOSTL(4)                TYPE C  ,             Cost center
          WERKS(4)                TYPE C  ,           " Plant
          MATNR(18)               TYPE C  ,             Material Code
          BWART(3)                TYPE C  ,           " Movement Type
          LGORT(4)                TYPE C  ,             Storage Location
          ERFMG(12)               TYPE C  ,             Quantity (TON)
          PRICE(11)               TYPE C  ,             Unit Price (USD)
          UOM(3)                  TYPE C  ,             Unit of measure
        END OF TYP_TMPDATA1.

TYPES : BEGIN OF TYP_MESSAGE,
          CATG(1)                 TYPE C  ,           " Category number
          SEQUE(2)                TYPE C  ,           " Sequence number
          MATNR(18)               TYPE C  ,           " Material Code
          BWART(3)                TYPE C  ,           " Movement Type
          KOSTL(4)                TYPE C  ,           " Cost center
          WERKS(4)                TYPE C  ,           " Plant
          LGORT(4)                TYPE C  ,           " Storage Location
          ERFMG(12)               TYPE C  ,           " Quantity
          PRICE(11)               TYPE C  ,           " Unit Price (USD)
          UOM(3)                  TYPE C  ,           " Unit of measure
          AMT_POST                LIKE VKDFS-NETWR, " Posting Amount(SGD)
          REMARK(200)             TYPE C  ,           " Remark
        END OF TYP_MESSAGE.

TYPES : BEGIN OF TYP_OBJ_NUM,
          MATNR(18)               TYPE C              " Material Code
          OBJ_NUM                 LIKE COSP-OBJNR ,   " Cost Center
        END OF TYP OBJ NUM.

TYPES : BEGIN OF TYP_VALUE,
          OBJNR                   LIKE ONR00-OBJNR,     Object number
          GJAHR                   LIKE COSP-GJAHR ,   " Year
          WRTTP                   LIKE COSP-WRTTP ,     Value type
          WOG001                  LIKE COSP-WOG001,     value currency
          WOG002                  LIKE COSP-WOG002,     value currency
          WOG003                  LIKE COSP-WOG003,     value currency
          WOG004                  LIKE COSP-WOG004,     value currency
          WOG005                  LIKE COSP-WOG005,     value currency
          WOG006                  LIKE COSP-WOG006,     value currency
          WOG007                  LIKE COSP-WOG007,     value currency
          WOG008                  LIKE COSP-WOG008,     value currency
          WOG009                  LIKE COSP-WOG009,   " value currency
          WOG010                  LIKE COSP-WOG010,     value currency
          WOG011                  LIKE COSP-WOG011,   " value currency
          WOG012                  LIKE COSP-WOG012,     value currency
          WOG013                  LIKE COSP-WOG013,   " value currency
          WOG014                  LIKE COSP-WOG014,   " value currency
          WOG015                  LIKE COSP-WOG015,   " value currency
          WOG016                  LIKE COSP-WOG016,   " value currency
        END OF TYP_VALUE.

TYPES : BEGIN OF TYP_TMP_AMT,
          OBJNR                   LIKE COEP-OBJNR,    " Object number
          GJAHR                   LIKE COEP-GJAHR,    " Year
          WRTTP                   LIKE COEP-WRTTP,    " Value type
          WKGBTR                  LIKE COEP-WKGBTR,   " Value CO currency
        END OF TYP_TMP_AMT.

TYPES : BEGIN OF TYP_TMP_AMT1,
          OBJNR                   LIKE COEP-OBJNR,    " Object number
          GJAHR                   LIKE COEP-GJAHR,    " Year
          WRTTP                   LIKE COEP-WRTTP,    " Value type
          WOGBTR                  LIKE COEP-WOGBTR,   " Value Object
        END OF TYP_TMP_AMT1.

TYPES : BEGIN OF TYP_TMP_AMT2,
          OBJNR                   LIKE COEP-OBJNR,    " Object number
          KSTAR                   LIKE COEP-KSTAR,    " Cost element
          GJAHR                   LIKE COEP-GJAHR,    " Year
          WRTTP                   LIKE COEP-WRTTP,    " Value type
          WKGBTR                  LIKE COEP-WKGBTR,   " Value CO currency
        END OF TYP_TMP_AMT2.

TYPES : BEGIN OF TYP_VALUE1,
          KOKRS                   LIKE COEPR-KOKRS ,  " Controlling Area
          OBJNR                   LIKE COEPR-OBJNR ,  " Object number
          SMEBTR                  LIKE COEPR-SMEBTR,  " quantity
        END OF TYP_VALUE1.

TYPES : BEGIN OF TYP_RET_MSG.
          INCLUDE STRUCTURE BDCMSGCOLL.
TYPES : END OF TYP_RET_MSG.

TYPES : BEGIN OF TYP_ZCPCSCOST,
          KSTAR                   LIKE COSP-KSTAR,    " Cost element
        END OF TYP_ZCPCSCOST.

TYPES : BEGIN OF TYP ZCPCSCOST2,
          KOSTL                   LIKE CSKS-KOSTL,      Cost center
          MATNR                   LIKE MARA-MATNR,      Material number
          KSTAR                   LIKE COSP-KSTAR,      Cost element
          STAGR                   LIKE COEPR-STAGR,     Statistical KEY
          COPDX                   TYPE C,               Co-product/single
          RATIO                   LIKE T5D7V-FAKTZ,     Ratio
        END OF TYP_ZCPCSCOST2.

* Internal table to get data from input file
DATA : I TMPDATA      TYPE TYP_TMPDATA   OCCURS  0  WITH HEADER LINE.

* Data category 1
DATA : I CAT1         TYPE TYP_TMPDATA   OCCURS  0  WITH HEADER LINE.

* Data category 2
DATA : I_CAT2         TYPE TYP_TMPDATA1  OCCURS  0  WITH HEADER LINE.
```

```
* Data category 3
DATA : I CAT3          TYPE TYP_TMPDATA   OCCURS  0  WITH HEADER LINE.

* Internal for keep error message
DATA : I ERROR         TYPE TYP MESSAGE   OCCURS  0  WITH HEADER LINE.

* Internal for keep success message
DATA : I SUCCESS       TYPE TYP MESSAGE   OCCURS  0  WITH HEADER LINE.

* Internal keep data for test mode
DATA : I TEST          TYPE TYP MESSAGE   OCCURS  0  WITH HEADER LINE.

* Internal table which message from call function
DATA : I RET MSG       TYPE TYP RET MSG   OCCURS  0  WITH HEADER LINE.

* BDC Internal table for posting BDC
DATA : I BDCDATA       LIKE BDCDATA       OCCURS  0  WITH HEADER LINE.

* Temporary table to keep object number for category 2
DATA : I OBJ NUM       TYPE TYP OBJ NUM   OCCURS 0  WITH HEADER LINE.

* Temporary table to keep object number for category 3
DATA : I OBJ NUM1      TYPE TYP OBJ NUM   OCCURS 0  WITH HEADER LINE.

* Temporary table to keep object number for category 3
DATA : I OBJ NON       TYPE TYP OBJ NUM   OCCURS 0  WITH HEADER LINE.

* Temporary value from select statement
DATA : I TMP VALUE  TYPE TYP TMP AMT      OCCURS 0  WITH HEADER LINE.

* Internal table to keep value for process
DATA : I VALUE         TYPE TYP TMP AMT   OCCURS 0  WITH HEADER LINE.

* Temporary amount from select statement
DATA : I TMP AMOUNT TYPE TYP TMP AMT      OCCURS 0  WITH HEADER LINE.

* Temporary quantity from select statement
DATA : I TMP QTY       TYPE TYP VALUE1    OCCURS 0 WITH HEADER LINE.

* Internal table to keep amount for process
DATA : I AMOUNT        TYPE TYP TMP AMT OCCURS 0 WITH HEADER LINE.

* Temporary table to keep amount of non steam material for process
DATA  :  I_TMP_AMOUNT_NON TYPE  TYP_TMP_AMT2  OCCURS  0   WITH HEADER
LINE.

* Internal table to keep amount of non steam material for process
DATA : I AMOUNT NON TYPE TYP TMP AMT2    OCCURS 0  WITH HEADER LINE.

*  internal table to keep data of non steam from  ZCPCSCOST2 for
category3
DATA : I ZCPCSCOST2 TYPE TYP ZCPCSCOST2 OCCURS  0 WITH HEADER LINE.

* Internal table to keep quantity for process
DATA : I QTY           TYPE TYP VALUE1    OCCURS 0  WITH HEADER LINE.

* Internal table to keep data from customize table ZCPCSCOST
DATA : I ZCPCSCOST  TYPE TYP ZCPCSCOST OCCURS 0  WITH HEADER LINE.

* Internal table keep date for post in category 2
```

```
DATA : I POST CAT2 TYPE TYP TMPDATA1 OCCURS 0 WITH HEADER LINE.

* Internal for category 2 for new calculate ratio
DATA : I CATEGORY2 TYPE TYP TMPDATA1 OCCURS 0 WITH HEADER LINE.

* Internal table to keep data from ZCPCSCOST3 (ratio for category 2)
DATA : I GET RATIO TYPE TYP ZCPCSCOST2 OCCURS 0 WITH HEADER LINE.

* - ----· Variable -----------------------------------.
DATA :
  V_MODE(1)           TYPE C                   Mode for BDC
  V_REMARK(200)       TYPE C                   Return message fr BDC
  V_YEAR              LIKE COSP-GJAHR          Year
  V_MONTH(2)          TYPE C                   Month
  ✓ BWART             LIKE RMO7M-BWARTWA       Movement type
  ✓ WERKS             LIKE RMO7M-WERKS         Plant
  V LGORT             LIKE RMO7M-LGORT         Storage location
  ✓_MATNR             LIKE MARA-MATNR          Material Number
  V_PRICE             TYPE F                   Price from file
  V_ERFMG             LIKE MSEG-ERFMG          Quantity from file
  ✓ POST_AMOUNT       LIKE MSEG-EXBWR          Amount to be post
  V TMP RATI          TYPE F                   Temporary ratio
                      TYPE F                   Get Ratio of Material 1
  ✓ TOTAL RATIO       TYPE F                   Total ratio
  V CAL_RATIO         TYPE F                 " Calculate ratio of Mat 1
  V_POST_             LIKE MSEG-EXBWR        Material line 1 to be post
  V_QTY_LINE1         LIKE MSEG-ERFMG        Quantity of Material line1
  ✓_TMP_QTY           LIKE MSEG-ERFMG          Quantity of Material
  V_CALQTY            LIKE MSEG-EXBWR          Quantity for calculate
  V_POSTCALQTY(17)    TYPE C                   Post Qty from calculate
  V_QTY(17)           TYPE C                   Quantity
  ✓ KOSTL             LIKE CSKS-KOSTL          Cost Center
  V_OBJNUM(16)        TYPE C                   Cost Center
  V_RATE              LIKE TCURR-UKURS         Exchange Rate
  ✓_REPID             LIKE SY-REPID            Report Id
  V_DOCDATE(8)        TYPE C                   Document Date
  ✓_POSTDATE(8)       TYPE C                   Posting Date
  V_STEAM_PRICE       LIKE MSEG-ERFMG          Unit price of steam
  ✓_POST_STEAM        LIKE MSEG-EXBWR          Steam Amount
  V_STEAM_QTY         LIKE COEPR-SMEBTR        Steam Quantity
  V_CAT1_BEFORE       TYPE I                   Error# before post
  V_CAT1 AFTER        TYPE I                   Error# after post
  V_UOM               LIKE MSEG-ERFME          Unit of measure
  ✓ PIS_TEXT          LIKE MKPF-BKTXT          PIS heade text

  - - - - - Range -----------------------------------
RANGES: R_COST        FOR COSP-KSTAR           Document DA to DZ
        R_OBJ_NUM1    FOR COEP-OBJNR           Object number
        R_OBJ_NUM2    FOR COEP-OBJNR           Object number
        R_ NON_STEAM  FOR COEP-MATNR           Material number
        R_COST_NON    FOR COEP-KSTAR           Cost element
        R_OBJ_NO      FOR COEP-OBJNR           Object non steam
        R_KOSTL       FOR ZCPCSCOST2-KOSTL,    Cost center
        R MATNR       FOR MARA-MATNR           Material number

**********************************************************************
       SELECTION SCREEN
*
**********************************************************************
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.
```

```
PARAMETERS:
  P_DATE LIKE COBK-BLDAT DEFAULT
              SY-DATUM                      " Posting date
  P_FILE LIKE RLGRAP-FILENAME DEFAULT
              C_DEFAULT_PATH OBLIGATORY.    " Input File

* Skip line.
SELECTION-SCREEN SKIP.

* Category 1 : Movement quantities of feed stock and by-product.
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS:
  RB CAT1  RADIOBUTTON GROUP R1.            " Category 1
SELECTION-SCREEN COMMENT 05(71) TEXT-004.
SELECTION-SCREEN END OF LINE.

* Skip line.
SELECTION-SCREEN SKIP.

* Category 2 : Finished Goods.
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS:
  RB_CAT2  RADIOBUTTON GROUP R1.            " Category 2
SELECTION-SCREEN COMMENT 05(71) TEXT-005.
SELECTION-SCREEN END OF LINE.

* Skip line.
SELECTION-SCREEN SKIP.

* Category 3 : Utility to Inventory.
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS:
  RB_CAT3  RADIOBUTTON GROUP R1.            " Category 3
SELECTION-SCREEN COMMENT 05(71) TEXT-006.
SELECTION-SCREEN END OF LINE.

* Factor values :
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 05(71) TEXT-003.
SELECTION-SCREEN END OF LINE.

* Steam Ultra High Pressure
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-007.
PARAMETERS:
  P_UHP(8) TYPE P DECIMALS 4 DEFAULT C_UHP.  " Steam Ultra High
Pressure
SELECTION-SCREEN END OF LINE.

* Steam High Pressure
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-008.
PARAMETERS:
  P_HP(8) TYPE P DECIMALS 4 DEFAULT C_HP.  " Stream High Pressure
SELECTION-SCREEN END OF LINE.

* Steam Medium Pressure
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-009.
PARAMETERS:
  P_MP(8) TYPE P DECIMALS 4 DEFAULT C_MP.   " Stream Medium Pressure
SELECTION-SCREEN END OF LINE.

* Steam Low Pressure
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-010.
PARAMETERS:
  P_LP(8) TYPE P DECIMALS 4 DEFAULT C_LP.   " Stream Low Pressure
SELECTION-SCREEN END OF LINE.

* Skip line.
SELECTION-SCREEN SKIP.

* Material Code of Steam SU
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-061.
PARAMETERS:
  P_SU     LIKE MARA-MATNR.                 " Steam SU
SELECTION-SCREEN END OF LINE.

* Material Code of Steam SH
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-062.
PARAMETERS:
  P_SH     LIKE MARA-MATNR.                 " Steam SH
SELECTION-SCREEN END OF LINE.

* Material Code of Steam SM
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-063.
PARAMETERS:
  P_SM     LIKE MARA-MATNR.                 " Steam SM
SELECTION-SCREEN END OF LINE.

* Material Code of Steam SL
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-064.
PARAMETERS:
  P_SL     LIKE MARA-MATNR.                 " Steam SL
SELECTION-SCREEN END OF LINE.

* Skip line.
SELECTION-SCREEN SKIP.

* Statistical Key Figure of Steam SU
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-065.
PARAMETERS:
  P_STAT LIKE COEPR-STAGR.                  " Statistical Key
SELECTION-SCREEN END OF LINE.

* Skip line.
SELECTION-SCREEN SKIP.

* Cost element of Allocation Steam SU
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 10(38) TEXT-066.
PARAMETERS:
  P COST LIKE COSP-KSTAR.                   " Cost Element
SELECTION-SCREEN END OF LINE.
```

```
* Skip line.
SELECTION-SCREEN SKIP.

SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS:
  P_TEST AS CHECKBOX default 'X'.           " Test mode check box
SELECTION-SCREEN COMMENT 5(9) TEXT-082.
SELECTION-SCREEN END OF LINE.

* Posting Mode : Display Error Only
* Mode For posting if P_mode = 'X', the BDC will be posted in
* Foreground
*  (error only) mode. if  P_MODE  = ",then post using normal
foreground.
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS:
  P_MODE AS CHECKBOX default ''.            BDC Mode'X'='E',''='A'
SELECTION-SCREEN COMMENT 5(32) TEXT-081.
SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN END OF BLOCK B1.

* Selection Block 2 : Material Definition
SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-002.
PARAMETERS:
  P_CONARE LIKE TKA01-KOKRS .                Controlling Area
SELECTION-SCREEN END OF BLOCK B2.

*********************************************************************
       INITIALIZATION
*
*********************************************************************
INITIALIZATION.

   V REPID = SY-REPID.

*********************************************************************
       TOP  OF  PAGE
*
*********************************************************************
TOP-OF-PAGE.

* Print top of page
  PERFORM F9300_TOP_OF_PAGE.

*********************************************************************
•      AT  SELECTION  SCREEN
*
*********************************************************************
AT SELECTION-SCREEN.

* Validation file and input screen
  PERFORM F1000_VALIDATION.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR P_FILE.

* Prepare default value of file name and path
  PERFORM F2000_PREPARE_DEFAULT_FILENAME.
```

```
**********************************************************************
       BEGIN    SELECTION
*
**********************************************************************
START-OF-SELECTION.

  CLEAR : V MODE.

* Upload records from input file into internal table and select
* difference from the cost center by using selection criteria from
* file
  PERFORM F3000_PREPARE_DATA.

* Main process for posting.
  PERFORM F4000_MAIN_PROCESS.

* Print report
  PERFORM F5000_PRINT_REPORT.

* Print End of report
  PERFORM F9000 END OF REPORT(ZREPEND) USING SY-LINSZ C_E.

END-OF-SELECTION.
**********************************************************************
       END  OF  SELECTION
*
**********************************************************************

**********************************************************************
       BEGIN    FORMS
*
**********************************************************************

**********************************************************************
*      FORM F1000_VALIDATION
                                                                     *
•      Description: This form is used for main validation.
**********************************************************************
FORM F1000 VALIDATION.

* Validate the file names exist.
  PERFORM F1100_CHECK_FILENAME_EXIST.

* Validate for input screen field.
  PERFORM F1200_CHECK_INPUT_SCREEN.

ENDFORM.                                  " 01000_VALIDATION

**********************************************************************
       FORM F1100 CHECK FILENAME EXIST

       Description: This form is used to validate the file names
                    exist. Validate for the file name and path,
                    Display error if the file doesn't exist.
**********************************************************************
FORM F1100_CHECK_FILENAME_EXIST.

* Validate for the file name and path and display error if the file
* doesn't exist.
  CALL FUNCTION 'WS QUERY'
```

```
       EXPORTING
           FILENAME        = P_FILE
           QUERY           = C_FE
       EXCEPTIONS
           INV_QUERY       = 1
           NO_BATCH        = 2
           FRONTEND_ERROR = 3
           OTHERS          = 4.

* Display error message if source file does not exist.
  IF SY-SUBRC <> 0.

* Source file does not exist.
    MESSAGE E000(ZY) WITH 'Source file does not exist!'(023).

  ENDIF.

ENDFORM.                          " F1100_CHECK_FILENAME_EXIST

*********************************************************************
       FORM F1200 CHECK INPUT SCREEN

•      Description: This form is used to validate input data from
                    screen.
*********************************************************************
FORM F1200 CHECK INPUT SCREEN.

* Checked Categories 1
  IF RB CAT1 = C X.

* Validate for category 1
    PERFORM F1210_VALIDATE_CATEGORY1.

* Checked Categories 2
  ELSEIF RB_CAT2 = C_X.

* Validate for category 2
    PERFORM F1220 VALIDATE CATEGORY2.

* Checked Categories 3
  ELSEIF RB CAT3 = C_X.

* Validate for category 3
    PERFORM F1230 VALIDATE_CATEGORY3.

  ENDIF.  " End check categories 1,2,3

  WRITE : P_DATE TO V_DOCDATE ,
          P_DATE TO V_POSTDATE.

ENDFORM.                          " F1200_CHECK_INPUT_SCREEN

*********************************************************************
•      FORM F1210_VALIDATE_CATEGORY1
*------------------------------------------------------------------
•      Description: This form is used to clear non related value
                    with category1.
*********************************************************************
FORM F1210 VALIDATE_CATEGORY1.
```

```
  CLEAR : P_SU
          P_SH
          P_SM
          P_SL
          P_CONARE
ENDFORM.                          " F1210 VALIDATE CATEGORY1

*********************************************************************
       FORM F1220_VALIDATE_CATEGORY2
*                                                                   *
       Description: This form is used validate related value with
                    category2.
*********************************************************************
FORM F1220_VALIDATE_CATEGORY2.

* Validate Controlling Area
  IF P_CONARE IS INITIAL.
    SET CURSOR FIELD 'P CONARE'.

* Please! Input Controlling Area before process
    MESSAGE E000(ZY) WITH
      'Please! Input Controlling Area before process'(039).

  ELSE.
    SELECT SINGLE KOKRS INTO TKA01-KOKRS
      FROM TKA01
     WHERE KOKRS = P CONARE.

    IF SY-SUBRC <> 0.
      SET CURSOR FIELD 'P CONARE'.

* Controlling Area not found!
      MESSAGE E000(ZY) WITH
        'Controlling Area not found!'(040).

    ENDIF.
  ENDIF.

ENDFORM.                          " F1220 VALIDATE CATEGORY2

*********************************************************************
       FORM F1230_VALIDATE_CATEGORY3
*                                                                   *
       Description: This form is used validate related value with
                    category3.
*********************************************************************
FORM F1230 VALIDATE CATEGORY3.

* Validate Steam Ultra High Pressure
  IF P_UHP IS INITIAL.
    SET CURSOR FIELD 'P UHP'.

* Please! Input Steam Ultra High Pressure before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam Ultra High Pressure before process'(031).

* Validate Steam High Pressure
  ELSEIF P_HP IS INITIAL.
    SET CURSOR FIELD 'P HP'.
```

```
* Please! Input Steam High Pressure before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam High Pressure before process'(032).

* Validate Steam Medium Pressure
  ELSEIF P_MP IS INITIAL.
    SET CURSOR FIELD 'P MP'.

* Please! Input Steam Medium Pressure before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam Medium Pressure before process'(033).

* Validate Steam Low Pressure
  ELSEIF P_LP IS INITIAL.
    SET CURSOR FIELD 'P LP'.

* Please! Input Steam Low Pressure before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam Low Pressure before process'(034).

  ENDIF.

* Validate Steam SU
  IF P_SU IS INITIAL.
    SET CURSOR FIELD 'P SU'.

* Please! Input Steam SU before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam SU before process'(035).

  ELSE.
    SELECT SINGLE MATNR INTO MARA-MATNR
      FROM MARA
     WHERE MATNR = P SU.

    IF SY-SUBRC <> 0.
      SET CURSOR FIELD 'P SU'.

* Material number not found!
      MESSAGE E000(ZY) WITH
        'Material number not found!'(029).

    ENDIF.
  ENDIF.

* Validate Steam SH
  IF P_SH IS INITIAL.
    SET CURSOR FIELD 'P SH'.

* Please! Input Steam SH before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam SH before process'(036).

  ELSE.
    SELECT SINGLE MATNR INTO MARA-MATNR
      FROM MARA
     WHERE MATNR = P SH.

    IF SY-SUBRC <> 0.
      SET CURSOR FIELD 'P SH'.
```

```
* Material number not found!
      MESSAGE E000(ZY) WITH
        'Material number not found!'(029).

    ELSE.

* Check material should not same as P_SU
      IF P SH = P SU.

        SET CURSOR FIELD 'P SH'.

* Material steam SH should not same as steam SU!
        MESSAGE E000(ZY) WITH
          'Material steam SH should not same as other steam!'(056).

      ENDIF.
    ENDIF.
  ENDIF.

* Validate Steam SM
  IF P_SM IS INITIAL.
    SET CURSOR FIELD 'P SM'.

* Please! Input Steam SM before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam SM before process'(037).

  ELSE.
    SELECT SINGLE MATNR INTO MARA-MATNR
      FROM MARA
     WHERE MATNR = P SM.

    IF SY-SUBRC <> 0.
      SET CURSOR FIELD 'P_SM'.

* Material number not found!
      MESSAGE E000(ZY) WITH
        'Material number not found!'(029).

    ELSE.

* Check material should not same as P_SU, P_SH
      IF P_SM = P_SU OR
         P SM = P SH.

        SET CURSOR FIELD 'P SM'.

* Material steam SM should not same as other steam!
        MESSAGE E000(ZY) WITH
          'Material steam SM should not same as other steam!'(057).

      ENDIF.
    ENDIF.
  ENDIF.

* Validate Steam SL
  IF P_SL IS INITIAL.
    SET CURSOR FIELD 'P SL'.
```

```
* Please! Input Steam SL before process
    MESSAGE E000(ZY) WITH
      'Please! Input Steam SL before process'(038).

  ELSE.
    SELECT SINGLE MATNR INTO MARA-MATNR
      FROM MARA
     WHERE MATNR = P_SL.

    IF SY-SUBRC <> 0.
      SET CURSOR FIELD 'P SL'.

* Please! Input Steam SL before process
      MESSAGE E000(ZY) WITH
        'Material number not found!'(029).

    ELSE.

* Check material should not same as P_SU, P_SH, P_SM
      IF P_SL = P_SU OR
         P_SL = P_SH OR
         P SL = P_SM.

        SET CURSOR FIELD 'P SL'.

* Material steam SL should not same as other steam!
        MESSAGE E000(ZY) WITH
          'Material steam SL should not same as other steam!'(058).

      ENDIF.
    ENDIF.
  ENDIF.

* Check statistical Key Figure of Steam SU
  IF P_STAT IS INITIAL.
    SET CURSOR FIELD 'P_STAT'.

* Please! Input Statistical Key Figure before process
    MESSAGE E000(ZY) WITH
      'Please! Input Statistical Key Figure before process'(068).
  ELSE.
    SELECT * UP TO 1 ROWS
      FROM TKA03
     WHERE STAGR = P_STAT.
    ENDSELECT.

    IF SY-SUBRC <> 0.
      SET CURSOR FIELD 'P_STAT'.

* Statistical Key Figure not found!
      MESSAGE E000(ZY) WITH
        'Statistical Key Figure not found!'(067).
    ENDIF.
  ENDIF.

* Validate Cost Element of Allocation Steam SU
  IF P_COST IS INITIAL.
    SET CURSOR FIELD 'P COST'.

* Please! Input Cost Element before process
```

```
  MESSAGE E000(ZY) WITH
    'Please! Input Cost Element before process'(069).
 ELSE.
  SELECT * UP TO 1 ROWS
    FROM CSKA
   WHERE KSTAR = P_COST.
  ENDSELECT.

  IF SY-SUBRC <> 0.
    SET CURSOR FIELD 'P_COST'.

* Cost Element not found!
    MESSAGE E000(ZY) WITH
      'Cost Element not found!'(070).
  ENDIF.
 ENDIF.

* Validate Controlling Area
 IF P_CONARE IS INITIAL.
  SET CURSOR FIELD 'P CONARE'.

* Please! Input Controlling Area before process
  MESSAGE E000(ZY) WITH
    'Please! Input Controlling Area before process'(039).

 ELSE.
  SELECT SINGLE KOKRS INTO TKA01-KOKRS
    FROM TKA01
   WHERE KOKRS = P CONARE.

  IF SY-SUBRC <> 0.
    SET CURSOR FIELD 'P CONARE'.

* Controlling Area not found!
    MESSAGE E000(ZY) WITH
      'Controlling Area not found!'(040).
  ENDIF.
 ENDIF.

ENDFORM.                    " F1230 VALIDATE CATEGORY3

*********************************************************************
       FORM F2000 PREPARE DEFAULT FILENAME

       Description: This form will enable the file selection option.*
*********************************************************************
FORM F2000 PREPARE DEFAULT FILENAME.

* Get file name from input selection.
 CALL FUNCTION 'WS_FILENAME_GET'
      EXPORTING
          MASK               = C_MASK
      IMPORTING
          FILENAME           = P_FILE
      EXCEPTIONS
          INV_WINSYS         = 1
          NO_BATCH           = 2
          SELECTION_CANCEL   = 3
          SELECTION_ ERROR   = 4
          OTHERS             = 5.
 IF SY-SUBRC <> 0.

* Unable to get file!
  MESSAGE E000(ZY) WITH 'Unable to get file!'(055).

 ENDIF.

ENDFORM.                          " F2000 PREPARE DEFAULT FILENAME

*********************************************************************
      FORM F3000 PREPARE DATA

      Description: This form will upload data from input file and  *
                   categorize each data record into internal table *
                   and get difference from the cost center by using*
                   cost center from file select from COSP table.
*********************************************************************
FORM F3000_PREPARE_DATA.

* Check Mode for posting if P_MODE = 'X' --> 'E', else MODE = 'A'.
  IF P_MODE = C_X.
    V_MODE = C_E.                 " Foreground Error Only Mode
  ELSE.
    V_MODE =                      " Foreground Mode
  ENDIF.

* Upload data from input file into internal table
  PERFORM F3100_UPLOAD_FILE.

* Select more data if have data to process
  IF NOT I_CAT1[] IS INITIAL OR
     NOT I_CAT2[] IS INITIAL OR
     NOT I CAT3[] IS INITIAL.

    IF NOT I_CAT2[] IS INITIAL AND
       RB_CAT2 = C_X.

* Select total value from COEP
      PERFORM F3200_GET_COEP.

* Get co-product/Single product and Ratio from ZCPCSCOST2
      PERFORM F3700_GET_RATIO.

    ENDIF.

    IF NOT I_CAT1[] IS INITIAL OR
       NOT I_CAT2[] IS INITIAL OR
       NOT I_CAT3[] IS INITIAL.

* Get Exchange Rate from TCURR
      PERFORM F3300_GET_EXCHANGE_RATE.

      IF NOT I_CAT3[] IS INITIAL AND
         RB CAT3 = C X.

* Get Amount of posted steam (COEP)
        PERFORM F3400_GET_AMOUNT_STEAM.

* Get Quantity of steam from stat key figures (COEPR)
```

```
        PERFORM F3500 GET QTY STEAM.

* Get Amount of posted non steam material
        PERFORM F3600_GET_AMOUNT_NON_STEAM.

      ENDIF.
    ENDIF.
  ENDIF.

ENDFORM.                           " F3000 PREPARE DATA

**********************************************************************
       FORM F3100_UPLOAD_FILE
*                                                                    *
       Description: This form will upload data from input file and  *
                    categorize each data record into internal table.*
**********************************************************************
FORM F3100_UPLOAD_FILE.

  REFRESH : I_TMPDATA
            I_CAT1
            I_CAT2
            I_CAT3
            I_ERROR
            I_SUCCESS
            I_TEST
            I_OBJ_NUM
            I_OBJ_NUM1
            R_OBJ_NUM1
            R_OBJ_NUM2
            I_OBJ_NON
            R_OBJ_NON
            R_KOSTL
            R MATNR
            I CATEGORY2

  CLEAR : I_TMPDATA
          I_CAT1
          I_CAT2
          I_CAT3
          I_ERROR
          I_SUCCESS
          I_TEST
          I_OBJ_NUM
          I_OBJ_NUM1
          R_OBJ_NUM1
          R_OBJ_NUM2
          I OBJ_NON
          R_OBJ_NON

          R MATNR
          I CATEGORY2

  V_YEAR  = P_DATE+0(4).
  V_MONTH = P DATE+4(2).

* Upload data file from file
  CALL FUNCTION 'UPLOAD'
       EXPORTING
           FILENAME  = P FILE
      FILETYPE  =  C_FILETYPE
    TABLES
      DATA_TAB  =  I_TMPDATA
    EXCEPTIONS
      CONVERSION_ ERROR         = 1
      INVALID_TABLE_WIDTH       = 2
      INVALID_TYPE              = 3
      NO_BATCH                  = 4
      UNKNOWN_ERROR             = 5
      GUI_REFUSE_FILETRANSFER   = 6
      OTHERS                    = 7.

 IF SY-SUBRC <> 0.

* Unable to upload file
  MESSAGE E000(ZY) WITH 'Unable to upload file'(022).

 ENDIF.

* Split each line into corresponding tables.
 LOOP AT I_TMPDATA.

  CASE I TMPDATA-CATG.

   WHEN '1'.
* Categories 1.
    MOVE : I_TMPDATA-CATG  TO I_CAT1-CATG
           I_TMPDATA-SEQUE TO I_CAT1-SEQUE
           I_TMPDATA-MATNR TO I_CAT1-MATNR
           I_TMPDATA-BWART TO I_CAT1-BWART
           I_TMPDATA-KOSTL TO I CAT1-KOSTL
           I_TMPDATA-WERKS TO I_CAT1-WERKS
           I_TMPDATA-LGORT TO I CAT1-LGORT
           I_TMPDATA-ERFMG TO I_CAT1-ERFMG
           I_TMPDATA-PRICE TO I_CAT1-PRICE
           I TMPDATA-UOM   TO I CAT1-UOM

    APPEND I_CAT1.
    CLEAR  I CAT1.

   WHEN '2'.
* Categories 2.
    MOVE : I_TMPDATA-CATG  TO I_CAT2-CATG
           I_TMPDATA-SEQUE TO I_CAT2-SEQUE
           I_TMPDATA-MATNR TO I_CAT2-MATNR
           I_TMPDATA-BWART TO I_CAT2-BWART
           I_TMPDATA-KOSTL TO I_CAT2-KOSTL
           I_TMPDATA-WERKS TO I_CAT2-WERKS
           I TMPDATA-LGORT TO I_CAT2-LGORT
           I_TMPDATA-ERFMG TO I_CAT2-ERFMG
           I_TMPDATA-PRICE TO I_CAT2-PRICE
           I_TMPDATA-UOM   TO I_CAT2-UOM

    APPEND  I_CAT2.
    CLEAR : I CAT2.

    MOVE : I_TMPDATA-CATG  TO I_CATEGORY2-CATG
           I_TMPDATA-SEQUE TO I_CATEGORY2-SEQUE
           I_TMPDATA-MATNR TO I_CATEGORY2-MATNR
           I TMPDATA-BWART TO I CATEGORY2-BWART
```

```
               I_TMPDATA-KOSTL TO I CATEGORY2-KOSTL
               I_TMPDATA-WERKS TO I CATEGORY2-WERKS
               I_TMPDATA-LGORT TO I CATEGORY2-LGORT
               I_TMPDATA-ERFMG TO I CATEGORY2-ERFMG
               I_TMPDATA-PRICE TO I CATEGORY2-PRICE
               I TMPDATA-UOM  TO I CATEGORY2-UOM

        APPEND  I_CATEGORY2.
        CLEAR : I CATEGORY2.

        CLEAR : V OBJNUM.

        V_OBJNUM+0(2) = C_KS  .
        V_OBJNUM+2(4) = P_CONARE+0(4).
        V_OBJNUM+6(6) = C_ZERO.
        V OBJNUM+12(4) = ITMPDATA-KOSTL.

        R_KOSTL-SIGN   = C_I.
        R_KOSTL-OPTION = C_EQ.
        R KOSTL-LOW    = I TMPDATA-KOSTL.

        APPEND R_KOSTL.
        CLEAR  R KOSTL.

        R_MATNR-SIGN   = C_I.
        R_MATNR-OPTION = C_EQ.
        R MATNR-LOW    = I TMPDATA-MATNR.

        APPEND R_MATNR.
        CLEAR  R_MATNR.

        READ TABLE I OBJ NUM WITH KEY OBJ NUM = V OBJNUM.

        IF SY-SUBRC <> 0.
          MOVE : I_TMPDATA-MATNR TO I_OBJ_NUM-MATNR  ,
                 V OBJNUM        TO I OBJ NUM-OBJ NUM.

          R_OBJ_NUM2-SIGN     = C_I.
          R_OBJ_NUM2-OPTION   = C_EQ.
          R OBJ NUM2-LOW      = V OBJNUM.

          APPEND R_OBJ_NUM2.
          CLEAR  R OBJ NUM2.

          APPEND I_OBJ_NUM.
          CLEAR  I_OBJ_NUM.
        ENDIF.

      WHEN '3'
* Categories 3
        MOVE : I_TMPDATA-CATG  TO I_CAT3-CATG
               I_TMPDATA-SEQUE TO I_CAT3-SEQUE
               I_TMPDATA-MATNR TO I_CAT3-MATNR
               I_TMPDATA-BWART TO I_CAT3-BWART
               I TMPDATA-KOSTL TO I_CAT3-KOSTL
               I_TMPDATA-WERKS TO I_CAT3-WERKS
               I_TMPDATA-LGORT TO I_CAT3-LGORT
               I_TMPDATA-ERFMG TO I_CAT3-ERFMG
               I_TMPDATA-PRICE TO I_CAT3-PRICE
               I TMPDATA-UOM   TO I_CAT3-UOM
        APPEND I_CAT3.
        CLEAR  I CAT3.

        IF I_TMPDATA-MATNR <> P_SU AND
           I_TMPDATA-MATNR <> P_SH AND
           I_TMPDATA-MATNR <> P_SM AND
           I TMPDATA-MATNR <> P SL.

* Non steam material
          R_NON_STEAM-SIGN   = C_I.
          R_NON_STEAM-OPTION = C_EQ.
          R NON STEAM-LOW    = I TMPDATA-MATNR.

          APPEND R_NON_STEAM.
          CLEAR  R NON STEAM.

          CLEAR : V OBJNUM.

          V_OBJNUM+0(2)   = C_KS .
          V_OBJNUM+2(4)   = P_CONARE+0(4).
          V_OBJNUM+6(6)   = C_ZERO.
          ✓ OBJNUM+12(4)  = I TMPDATA-KOSTL.

          READ TABLE I OBJ NON WITH KEY OBJ NUM = V OBJNUM.

          IF SY-SUBRC <> 0.
            MOVE : I_TMPDATA-MATNR TO I_OBJ_NON-MATNR   ,
                   V OBJNUM        TO I OBJ NON-OBJ NUM.

            R_OBJ_NON-SIGN     = C_I.
            R_OBJ_NON-OPTION   = C_EQ.
            R OBJ NON-LOW      = V OBJNUM.

            APPEND R_OBJ_NON.
            CLEAR R OBJ NON.

            APPEND I_OBJ_NON.
            CLEAR  I_OBJ_NON.
          ENDIF.

        ELSE.

* Steam material
          CLEAR : V OBJNUM.

          ✓ OBJNUM+0(2)   = C_KS
          V_OBJNUM+2(4)   = P_CONARE+0(4).
          V_OBJNUM+6(6)   = C_ZERO.
          ✓ OBJNUM+12(4)  = I TMPDATA-KOSTL.

          READ TABLE I OBJ NUM1 WITH KEY OBJ NUM = V OBJNUM.

          IF SY-SUBRC <> 0.
            MOVE : I_TMPDATA-MATNR TO I_OBJ_NUM1-MATNR   ,
                   V OBJNUM        TO I OBJ NUM1-OBJ NUM.

            R_OBJ_NUM1-SIGN     = C_I.
            R_OBJ_NUM1-OPTION   = C_EQ.
            R OBJ NUM1-LOW      = V OBJNUM.
            APPEND R_OBJ_NUM1.
            CLEAR  R OBJ NUM1.

            APPEND I_OBJ_NUM1.
            CLEAR  I_OBJ_NUM1.
          ENDIF.
        ENDIF.
    ENDCASE.  " Case check categories 1, 2 and 3

* Validate UOM from file : if blank will not process and raise error.
    IF I_TMPDATA-UOM IS INITIAL OR
      I TMPDATA-UOM = ''.

      MOVE : I_TMPDATA-CATG  TO I_ERROR-CATG
             I_TMPDATA-SEQUE TO I_ERROR-SEQUE
             I_TMPDATA-MATNR TO I_ERROR-MATNR
             I_TMPDATA-BWART TO I_ERROR-BWART
             I_TMPDATA-KOSTL TO I_ERROR-KOSTL
             I_TMPDATA-WERKS TO I_ERROR-WERKS
             I TMPDATA-LGORT TO I_ERROR-LGORT
             I_TMPDATA-ERFMG TO I_ERROR-ERFMG
             I_TMPDATA-PRICE TO I_ERROR-PRICE
             I TMPDATA-UOM  TO I ERROR-UOM

      I ERROR-AMT POST = 0.

* Exchange Rate Not Found!
      MOVE : 'Unit of Measure Not Found!'(083) TO I ERROR-REMARK.

      APPEND I_ERROR.
      CLEAR  I_ERROR.
    ENDIF.

  ENDLOOP.  " End loop I_TMPDATA

ENDFORM.                          " F3100 UPLOAD FILE

*********************************************************************
*      FORM F3200_GET_COEP                                          *
*
*      Description: This form will get difference from CO Object
*                   (COEP) and cost center should be in customize
*                   ZCPCSCPST, Ledger (LEDNR) = 00, Year (GJAHR) =  *
*                   posting year, Value type (WRTTP) = 04, Version
*                   (VERSN) = 000.
*********************************************************************
FORM F3200_GET_COEP.

  REFRESH : I_TMP_VALUE ,
            I_VALUE
            I_ZCPCSCOST ,
            R COST

  CLEAR : I_TMP_VALUE
          I_VALUE
          I_ZCPCSCOST
          R COST

  SELECT KSTAR
```

```
    FROM ZCPCSCOST
    INTO TABLE I_ZCPCSCOST.

  LOOP AT I_ZCPCSCOST.

    R_COST-SIGN    = C_I.
    R_COST-OPTION  = C_EQ.
    R COST-LOW     = I ZCPCSCOST-KSTAR.

    APPEND R_COST.
    CLEAR  R COST.

  ENDLOOP.

* Check internal table before using for all entries statement
  IF NOT I OBJ NUM[] IS INITIAL.

    SELECT OBJNR         " Object number
           GJAHR         " Year
           WRTTP         " Value type
           WKGBTR        " Total value in CO currency
      FROM COEP
      INTO TABLE I_TMP_VALUE
     WHERE LEDNR = C_00
       AND OBJNR IN R_OBJ_NUM2
       AND GJAHR = V_YEAR
       AND WRTTP = C_04
       AND VERSN = C_000
       AND KSTAR IN R_COST
       AND PERIO = V_MONTH.

    IF SY-SUBRC = 0.
      LOOP AT I_TMP_VALUE.

* Insert information for calculate difference from Cost Center
        MOVE : I TMP VALUE TO I VALUE.

        COLLECT I_VALUE.
        CLEAR   I VALUE.

      ENDLOOP.
    ELSE.

* Total value not found in object currency!
      MOVE : 'Total value not found in object currency!'(050)
             TO I ERROR-REMARK.

      APPEND I_ERROR.
      CLEAR I ERROR.

    ENDIF.
  ENDIF.  " End if NOT I OBJ NUM[] IS INITIAL.

ENDFORM.                            " F3200_GET_COEP

*********************************************************************
       FORM F3300 GET EXCHANGE RATE

•      Description: This form will get exchange from exchange rate  *
                    table (TCURR) and input from USD currency to
                    SGD currency.
*********************************************************************
FORM F3300_GET_EXCHANGE_RATE.

  CLEAR : V RATE.

* Call function to find exchange rate with type = M
  CALL FUNCTION 'READ_EXCHANGE_RATE
    EXPORTING
      CLIENT                  = SY-MANDT
      DATE                    = P_DATE
      FOREIGN_ CURRENCY       = C_USD
      LOCAL_CURRENCY          = C_SGD
      TYPE_OF_RATE            = CEXGTYP
   IMPORTING
      EXCHANGE_RATE           = V_RATE.

ENDFORM.                          " F3300 GET EXCHANGE RATE

*********************************************************************
       FORM F3400 GET AMOUNT STEAM                                  *

       Description: This form will get amount of posted steam
                    (COEP-WKGBTR) from CO Object: Line Items
                    (by Period) with criteria are Ledger (LEDNR)=0, *
                    Year (GJAHR) = posting year, Value type (WRTTP)=*
                    04, Version (VERSN) = 000 a& cost element(KSTAR)*
                    = Selection screen, period (PERIO) = post month *
*********************************************************************
FORM F3400_GET_AMOUNT_STEAM.

  REFRESH : I_TMP_AMOUNT,
            I AMOUN

  CLEAR : I_TMP_AMOUNT  ,
          I AMOUNT

* Check internal table before using for all entries statement
  IF NOT I_OBJ_NUM1[] IS INITIAL.
    SELECT OBJNR        " Object number
           GJAHR        " Year
           WRTTP        " Value type
           WKGBTR       " Total value in CO currency
      FROM COEP
      INTO TABLE I_TMP_AMOUNT
     WHERE LEDNR = C_00
       AND OBJNR IN R_OBJ_NUM1
       AND GJAHR = V_YEAR
       AND WRTTP = C_04
       AND VERSN = C_000
       AND KSTAR = P_COST
       AND PERIO = V MONTH.

    IF SY-SUBRC = 0.
      LOOP AT I_TMP_AMOUNT.

* Insert information for amount steam
        MOVE : I TMP AMOUNT TO I AMOUNT.

        COLLECT I_AMOUNT.
        CLEAR   I_AMOUNT.

      ENDLOOP.
    ELSE.

* Amount not found for post steam!
      MOVE : 'Amount not found for post steam!'(051)
             TO I ERROR-REMARK.

      APPEND I_ERROR.
      CLEAR I ERROR.

    ENDIF.
  ENDIF.  " End if NOT I OBJ NUM1[] IS INITIAL.

ENDFORM.                          " F3400 GET AMOUNT STEAM

**********************************************************************
       FORM F3500 GET QTY STEAM

       Description: This form will get quantity of posted steam
                    (COEPR-SMEBTR) from CO Object:Items for Stat.
                    Key Figs (by Period) (COEPR) with criteria are  *
                    Ledger (LEDNR) = 00, year (GJAHR)=posting year,
                    Value type (WRTTP) = 04, Version (VERSN) = 000,
                    Steam Statistical Key Figure (STAGR) = U01 and
                    Period (PERIO) = posting month.
**********************************************************************
FORM F3500 GET QTY STEAM.

  REFRESH : I_TMP_QTY,
            I QTY

  CLEAR : I_TMP_QTY  ,
          I QTY

* Check internal table before using for all entries statement
  IF NOT I_OBJ_NUM1 IS INITIAL.
    SELECT KOKRS          " Controlling Area
           OBJNR          " Object number
           SMEBTR         " Statistical quantity
      FROM COEPR
      INTO TABLE I_TMP_QTY
     WHERE LEDNR = C_00
       AND OBJNR IN R_OBJ_NUM1
       AND GJAHR = V_YEAR
       AND WRTTP = C_04
       AND VERSN = C_000
       AND STAGR = P_STAT
       AND PERIO = V_MONTH.

    IF SY-SUBRC = 0.
      LOOP AT I TMP QTY.

* Insert information for calculate difference from Cost Center
        MOVE : I TMP QTY TO I QTY.

        COLLECT I_QTY.
        CLEAR   I QTY.
      ENDLOOP.
    ELSE.

* Amount not found for post steam!
      MOVE : 'Amount not found for post steam!'(051)
             TO I ERROR-REMARK.

      APPEND I_ERROR.
      CLEAR I ERROR.

    ENDIF.
  ENDIF.  " End if NOT I OBJ NUM1[] IS INITIAL.

ENDFORM.                           " F3500_GET_QTY_STEAM

**********************************************************************
       FORM F3600 GET AMOUNT NON STEAM

       Description: This form will get amount of posted steam
                    (COEP-WKGBTR) from CO Object: Line Items
                    (by Period) with criteria are Ledger (LEDNR)=0, *
                    Year (GJAHR)=posting year, Value type (WRTTP)=
                    04, Version (VERSN) = 000 & cost element (KSTAR)*
                    = Custom table related with non steam material, *
                    period (PERIO) = post month
**********************************************************************
FORM F3600_GET_AMOUNT_NON_STEAM.

  REFRESH : I_TMP_AMOUNT_NON,
            I_AMOUNT_NON
            I_ZCPCSCOST2
            R COST NON

  CLEAR : I_TMP_AMOUNT_NON ,
          I_AMONN   ON
          I_ZCPCSCOST2
          R COST NON

   SELECT KOSTL               Cost center
          MATNR               Material number
          KSTAR               Cost element
          STAGR               Statistical KEY
          COPDX               Cost element
          RATIO               Material number
    FROM ZCPCSCOST2
    INTO TABLE I_ZCPCSCOST2
   WHERE MATNR IN R NON STEAM.

  IF SY-SUBRC <> 0.
    MOVE : 'Not found Cost Element in table ZCPCSCOST2!'(085)
           TO I ERROR-REMARK.

    APPEND I_ERROR.
    CLEAR  I_ERROR.
  ELSE.

    LOOP AT I_ZCPCSCOST2.

      R_COST_NON-SIGN    = C_I.
      R COST NON-OPTION  = C_EQ.
      R COST NON-LOW      = I ZCPCSCOST2-KSTAR.

      APPEND R_COST_NON.
      CLEAR  R COST NON.

    ENDLOOP.
  ENDIF.

* Check internal table before using for all entries statement
  IF NOT I_OBJ_NUM1[] IS INITIAL.
    SELECT OBJNR         " Object number
           KSTAR         " Cost element
           GJAHR         " Year
           WRTTP         " Value type
           WKGBTR        " Total value in CO currency
      FROM COEP
      INTO TABLE I_TMP_AMOUNT_NON
     WHERE LEDNR = C_00
       AND OBJNR IN R_OBJ_NON
       AND GJAHR = V_YEAR
       AND WRTTP = C_04
       AND VERSN = C_000
       AND KSTAR IN R_COST_NON
       AND PERIO = V MONTH.

    IF SY-SUBRC = 0.
      LOOP AT I_TMP AMOUNT_ NON.

* Insert information for amount of non steam material
        MOVE : I TMP AMOUNT NON TO I AMOUNT NON.

        COLLECT I_AMOUNT_NON.
        CLEAR I AMOUNT NON.

      ENDLOOP.
    ELSE.

* Amount not found for post steam!
      MOVE : 'Amount not found for post non steam!'(086)
             TO I ERROR-REMARK.

      APPEND I_ERROR.
      CLEAR I ERROR.

    ENDIF.
  ENDIF.  " End if NOT I OBJ NUM1[] IS INITIAL.

ENDFORM.                          " F3600 GET AMOUNT NON STEAM

*********************************************************************
       FORM F3700 GET RATIO                                         *
                                                                    *
•      Description: This form use to get co product/Single product& *
                    Ratio for calcution in category 2.
*********************************************************************
FORM F3700_GET_RATIO.

  REFRESH : I GET RATIO.

  CLEAR   : I_GET_RATIO.
  SELECT KOSTL                   Cost center
         MATNR                   Material number
         KSTAR                   Cost element
         STAGR                   Statistical KEY
         COPDX                   Cost element
         RATIO                   Material number
   FROM ZCPCSCOST2
   INTO TABLE I_GET_RATIO
  WHERE KOSTL IN R_KOSTL
    AND MATNR IN R MATNR.

  IF SY-SUBRC = 0.
    SORT I_GET_RATIO BY KOSTL MATNR.
  ELSE.

* Not Found Co-Product/Single product and Ratio!
    MOVE : 'Not Found Co-Product/Single product and Ratio!'(087)
           TO I ERROR-REMARK.

    APPEND I_ERROR.
    CLEAR I ERROR.

  ENDIF.

ENDFORM.                          " F3700_GET_RATIO

*********************************************************************
       FORM F4000 MAIN PROCESS                                      *
                                                                    *
       Description: This form is main process.                      *
*********************************************************************
FORM F4000_MAIN_PROCESS.

* Main process for posting
  PERFORM F4100_POSTING_CATEGORY.

ENDFORM.                          " F4000 MAIN PROCESS

*********************************************************************
       FORM F4100_POSTING_CATEGORY                                  *
*                                                                   *
       Description: This form is main process.
*********************************************************************
FORM F4100_POSTING_CATEGORY.

*  Check for Categories 1 for posting via transaction  MB1A(Good
issues)
* and MB1C(Goods receipt)
  IF RB CAT1 = C X.

* Check I_CAT1 before posting , should have data before post
    IF NOT I CAT1[] IS INITIAL.

      CLEAR : V_PIS_TEXT.
      CONCATENATE C PIS TEXT C CHAR1 INTO V PIS TEXT.

      SORT I CAT1 BY CATG SEQUE MATNR BWART.

      LOOP AT I_CAT1.
```

```
* Movement Type for good issues = 901
        IF I CAT1-BWART = C 901.

          CLEAR : V_BWART ,
                  V_WERKS
                  V_LGORT
                  V_MATNR ,
                  V_QTY
                  V_KOSTL ,
                  V UOM

          MOVE : I_CAT1-BWART TO V_BWART,
                 I CAT1-WERKS TO V_WERKS,
                 I CAT1-LGORT TO V_LGORT,
                 I CAT1-MATNR TO V_MATNR,
                 I CAT1-ERFMG TO V_QTY   ,
                 I CAT1-KOSTL TO V_KOSTL,
                 I CAT1-UOM  TO V UOM   .

* Check test mode for posting
          IF P_TEST = ''.

            CLEAR : V_CAT1_BEFORE ,
                    V CAT1 AFTER  .

            DESCRIBE TABLE I_ERROR LINES V_CAT1_BEFORE.

* Posting BDC for Good Issues (MB1A).
            PERFORM F9100_GOOD_ISSUES TABLES I_CAT1
                                      USING  V_BWART V_WERKS V_LGORT
                                             V_MATNR V_QTY   V_KOSTL
                                             V UOM.

            DESCRIBE TABLE I_ERROR LINES V_CAT1_AFTER.

* If error after call transaction > before should terminate because
*  for category 1 should stop the processing of the rest of the
category.
            IF V_CAT1_AFTER > V_CAT1_BEFORE.
              EXIT.
            ENDIF.

          ELSE.

* Move data for print test mode
            MOVE : I_CAT1 TO I_TEST.
            I TEST-AMT POST = 0.

            APPEND I_TEST.
            CLEAR  I_TEST.
          ENDIF.

* Movement Type for good receipt = 521
        ELSEIF I CAT1-BWART = C 521.

          CLEAR : V_BWART ,
                  V_WERKS ,
                  V_LGORT ,
                  V_MATNR ,
                  V_PRICE ,
                  V_ERFMG ,
                  V_CALQTY,
                  V_QTY   ,
                  V_KOSTL ,
                  V_POSTCALQTY,
                  V UOM   .

* Check Exchange Rate
          IF V RATE IS INITIAL.

            MOVE : I_CAT1 TO I_ERROR.
            I ERROR-AMT POST = 0.

* Exchange Rate Not Found!
            MOVE : 'Exchange Rate Not Found!'(046) TO I ERROR-REMARK.

            APPEND I_ERROR.
            CLEAR  I_ERROR.
          ELSE.

            MOVE : I_CAT1-ERFMG TO V_ERFMG,
                   I CAT1-PRICE TO V_PRICE.

* Amount = Quantity * Unit Price (USD) * Exchange rate (USD => SGD)

            V CALQTY = V ERFMG * ( V PRICE * V RATE )

            MOVE : V CALQTY      TO V POSTCALQTY

            MOVE : I_CAT1-BWART  TO V_BWART ,
                   I_CAT1-WERKS  TO V_WERKS ,
                   I_CAT1-LGORT  TO V_LGORT ,
                   I_CAT1-MATNR  TO V_MATNR ,
                   I_CAT1-ERFMG  TO V_QTY   ,
                   I_CAT1-KOSTL  TO V_KOSTL ,
                   I CAT1-UOM     TO V UOM   .

* Check test mode for posting
            IF P_TEST = ''.

              CLEAR : V_CAT1_BEFORE ,
                      V CAT1 AFTER  .

              DESCRIBE TABLE I_ERROR LINES V_CAT1_BEFORE.

* Posting BDC for Good Receipt (MB1C).
              PERFORM F9200_GOOD_RECEIPT TABLES I_CAT1
                                         USING  V_BWART V_WERKS
                                                V_LGORT V_MATNR
                                                V_QTY   V_KOSTL
                                                V_POSTCALQTY
                                                V UOM.

              DESCRIBE TABLE I_ERROR LINES V_CAT1_AFTER.

* If error after call transaction > before should terminate because
*  for category 1 should stop the processing of the rest of the
category.
              IF V_CAT1_AFTER > V_CAT1_BEFORE.
```

```
                 EXIT.
               ENDIF.

             ELSE.

* Move data for print test mode
               MOVE : I_CAT1 TO I_TEST.
               MOVE : V_POSTCALQTY TO I_TEST-AMT_POST .

               APPEND I_TEST.
               CLEAR I TEST.

             ENDIF.  " End check test mode

           ENDIF.  " End check V RATE is initial

         ENDIF.  " End of check movement type 901,521

       ENDLOOP.  " End loop I_CAT1

     ENDIF.  " ENDIF NOT I CAT1[] IS INITIAL.

* Check for Categories 2 for posting via trans MB1C (Good Receipt)
  ELSEIF RB CAT2 = C X.

* Check I_CAT2 before posting , should have data before post
    IF NOT I CAT2[] IS INITIAL.

      REFRESH : I POST CAT2 .

      CLEAR : V_PIS_TEXT
              I POST CAT2

      CONCATENATE C PIS TEXT C CHAR2 INTO V PIS TEXT.

      SORT I CATEGORY2 BY CATG SEQUE KOSTL WERKS MATNR.

      LOOP AT I_CATEGORY2.

        CLEAR : V_TMP_RATIO  ,
                V TMP QTY

        READ TABLE I_GET_ RATIO WITH KEY KOSTL = I CATEGORY2-KOSTL
                                         MATNR = I CATEGORY2-MATNR.

        IF SY-SUBRC = 0.

          IF I GET RATIO-COPDX = C_C.

            MOVE : I_GET_RATIO-RATIO TO V_TMP_RATIO,
                   I CATEGORY2-ERFMG TO V TMP QTY  .

* Calculation total ratio for each cost center before posting
            V_TOTAL_RATIO = [ ( Quantity Mat.1 * Ratio mat.1 ) +
                              ( Quantity Mat.2 * Ratio mat.2 ) +

                              ( Quantity Mat.X * Ratio mat.X ) J.

            V TOTAL RATIO = V TOTAL RATIO + ( V TMP QTY * V TMP RATIO
            MOVE I CATEGORY2 TO I POST CAT2.

            APPEND I_POST_CAT2.
            CLEAR  I POST CAT2.

            AT END OF KOSTL.

              READ TABLE I_VALUE WITH KEY
                                 OBJNR+12(4) = I CATEGORY2-KOSTL.

              IF SY-SUBRC = 0.

* Calculation portion for ratio calculation
                PERFORM F4110 PORTION WITH RATIO.

              ELSE.

                MOVE-CORRESPONDING I CATEGORY2 TO I ERROR.

                I ERROR-AMT POST = 0.

* Cost center not found in cost table for posting ethylene!
                MOVE : 'Not Found Cost center in cost table!'(045)
                        TO I ERROR-REMARK.

              ENDIF.  " End read I VALUE

            ENDAT.

          ELSEIF I GET RATIO-COPDX = CS.

* Should post all remain data incase of calculate total and
             found that next record has COPDX = 'S', so all previous
             records should post for case of COPDX = 'C'.
            IF NOT I POST CAT2[] IS INITIAL.

* Calculation portion for ratio calculation
              PERFORM F4110_PORTION_WITH_RATIO.

            ENDIF.
* Normal posting without calculation ratio
            PERFORM F4120 POST WITHOUT RATIO.

          ENDIF.
        ELSE.
          MOVE-CORRESPONDING I CATEGORY2 TO I ERROR.

          I ERROR-AMT POST = 0.

* Not Found Co-Product/Single product and Ratio!
          MOVE   : 'Not Found Co-Product/Single product and
Ratio!'(087)
                 TO I ERROR-REMARK.

          APPEND I_ERROR.
          CLEAR  I_ERROR.
        ENDIF.
      ENDLOOP.  " End loop I_CATEGORY2
    ENDIF.  " End if NOT I CAT2[] IS INITIAL.
```

```
* Check for Categories 3 for posting via trans MB1C (Good Receipt)
  ELSEIF RB CAT3 = C X.

* Check I_CAT3 before posting , should have data before post
    IF NOT I CAT3[] IS INITIAL.

      CLEAR : V_PIS_TEXT.
      CONCATENATE C PIS TEXT C CHAR3 INTO V PIS TEXT.

      SORT I CAT3 BY CATG SEQUE MATNR BWART.

* Loop internal table in categories 3 for posting
      LOOP AT I_CAT3.

* Check steam material for categories 3
        IF I_CAT3-MATNR = P_SU OR
           I_CAT3-MATNR = P_SH OR
           I_CAT3-MATNR = P_SM OR
           I CAT3-MATNR = P_SL.

          CLEAR : V_POST_STEAM ,
                  V_STEAM_PRICE,
                  V_STEAM_QTY
                  V_POSTCALQTY ,
                  V_BWART
                  ✓ WERKS
                  V_LGORT
                  V_MATNR
                  V_QTY
                  V_KOSTL
                  ✓ UOM

* Get quantity from COEPR table
          READ TABLE I QTY WITH KEY OBJNR+12(4) = I CAT3-KOSTL.

          IF SY-SUBRC = 0.
            MOVE : I QTY-SMEBTR TO V STEAM QTY.

* Check V_STEAM_QTY before divide amount, should not divide by 0
            IF V STEAM QTY <> 0.

* Get amount from COEP table with posting month criteria
              LOOP AT I_AMOUNT WHERE OBJNR+12(4) = I_CAT3-KOSTL.
                V_STEAM_PRICE = I_AMOUNT-WKGBTR / V_STEAM_QTY.
              ENDLOOP.

* Posting Steam Material
* Check Steam Material for posting
* Case of steam material
              CASE I CAT3-MATNR.

* Calculate post steam amount in SGD = unit price of steam * quantity
*
* factor values * Exchange rate
                WHEN P_SU.
                  ✓ POST STEAM = V STEAM PRICE * I CAT3-ERFMG *
                                 P UHP * V RATE.

                WHEN P_SH.
```

```
                  V_POST_STEAM = V_STEAM_PRICE * I_CAT3-ERFMG *
                                 P_HP * V RATE.

                WHEN P_SM.
                  V_POST_STEAM = V_STEAM_PRICE * I_CAT3-ERFMG *
                                 P_MP * V RATE.

                WHEN P_SL.
                  V_POST_STEAM = V_STEAM_PRICE * I_CAT3-ERFMG *
                                 P LP * V_RATE.

              ENDCASE.

* Move values before posting
              MOVE : V_POST_STEAM TO V POSTCALQTY.

              MOVE : I_CAT3-BWART  TO V BWART
                     I_CAT3-WERKS  TO V_WERKS ,
                     I_CAT3-LGORT  TO V_LGORT ,
                     I_CAT3-MATNR  TO V_MATNR ,
                     I_CAT3-ERFMG  TO V_QTY   ,
                     I_CAT3-KOSTL  TO V_KOSTL ,
                     I CAT3-UOM     TO V UOM   .

* Check test mode before posting
              IF P_TEST =

* Posting BDC for Good Receipt (MB1C).
                PERFORM F9200_ GOOD_RECEIPT TABLES I_CAT3
                                           USING  V_BWART V_WERKS
                                                  V_LGORT V_MATNR
                                                  V_QTY   V_KOSTL
                                                  V_POSTCALQTY

              ELSE.
* Move data for print test mode
                MOVE : I_CAT3 TO I_TEST.
                MOVE : V POSTCALQTY TO I TEST-AMT POST .

                APPEND I_TEST.
                CLEAR I TEST.

              ENDIF.  " End check test mode

            ELSE.  " Else check if V STEAM QTY <> 0

              MOVE : I_CAT3 TO I_ERROR.
              I ERROR-AMT POST = 0.

* Steam quantity = 0
              MOVE  : 'Steam quantity is  Zero!'(054) TO  I_ERROR-
REMARK.

              APPEND I_ERROR.
              CLEAR I ERROR.

            ENDIF.  " End of if V STEAM QTY <> 0

          ELSE.  " Else sy-subrc = 0 of I_QTY
```

```
            MOVE : I_CAT3 TO I_ERROR.
            I ERROR-AMT POST = 0.

* Steam quantity not found!
            MOVE : 'Steam quantity not found!'(052)
                   TO I ERROR-REMARK.

            APPEND I_ERROR.
            CLEAR I ERROR.

          ENDIF.  " End of read table I QTY

        ELSE.  " else of check material for categories 3
* Other case for cat 3 using normal posting depend on Movement type

* Movement Type for good issues = 901
          IF I CAT3-BWART = C 901.

            CLEAR : V_BWART ,
                    V_WERKS ,
                    V_LGORT ,
                    V_MATNR ,
                    V_QTY
                    V_KOSTL ,
                    V UOM

            MOVE : I_CAT3-BWART TO V_BWART,
                   I_CAT3-WERKS TO V_WERKS,
                   I_CAT3-LGORT TO V_LGORT,
                   I_CAT3-MATNR TO V_MATNR,
                   I_CAT3-ERFMG TO V_QTY   ,
                   I_CAT3-KOSTL TO V_KOSTL,
                   I CAT3-UOM  TO V UOM   .

* Check test mode before posting
            IF P TEST = ''.

* Posting BDC for Good Issues (MB1A).
              PERFORM F9100_GOOD_ISSUES TABLES I_CAT3
                                        USING V_BWART V_WERKS V_LGORT
                                              V_MATNR V_QTY   V_KOSTL
                                              V UOM.
            ELSE.
* Move data for print test mode
              MOVE : I_CAT3 TO I_TEST.
              I TEST-AMT POST = 0.

              APPEND I_TEST.
              CLEAR  I_TEST.
            ENDIF.

* Movement Type for good receipt = 521
          ELSEIF I CAT3-BWART = C 521.

            CLEAR : V_BWART ,
                    V_WERKS ,
                    V_LGORT ,
                    V_MATNR ,
                    V_PRICE ,
                    V ERFMG ,
                    V_CALQTY,
                    V_QTY    ,
                    V_KOSTL ,
                    V_POSTCALQTY,
                    V UOM    .

* Check Exchange Rate
            IF V RATE IS INITIAL.

              MOVE : I_CAT3 TO I_ERROR.
              I ERROR-AMT POST = 0.

* Exchange Rate Not Found!
              MOVE  : 'Exchange Rate Not  Found!'(046) TO I ERROR-
REMARK.

              APPEND I_ERROR.
              CLEAR  I_ERROR.
            ELSE.

              MOVE : I_CAT3-ERFMG TO V_ERFMG,
                     I CAT3-PRICE TO V_PRICE.

* Posted amount in SGD = Difference from cost center USD * exchange
rate

              CLEAR : I_ZCPCSCOST2,
                      I AMOUNT NON.

              READ TABLE I ZCPCSCOST2 WITH KEY MATNR = I CAT3-MATNR.

              READ TABLE I_AMOUNT_NON
                WITH KEY OBJNR+12(4) = I CAT3-KOSTL
                         KSTAR       = I ZCPCSCOST2-KSTAR.

              IF SY-SUBRC = 0.
                V_POSTCALQTY = I_AMOUNT_NON-WKGBTR * V_RATE.
              ELSE.
                V_POSTCALQTY = 0.
              ENDIF.

              MOVE : I_CAT3-BWART TO V_BWART ,
                     I_CAT3-WERKS TO V_WERKS ,
                     I_CAT3-LGORT TO V_LGORT ,
                     I_CAT3-MATNR TO V_MATNR ,
                     I_CAT3-ERFMG TO V_QTY   ,
                     I_CAT3-KOSTL TO V_KOSTL ,
                     I CAT3-UOM  TO V UOM    .

* Check test mode before posting
              IF P_TEST = ''.

* Posting BDC for Good Receipt (MB1C).
                PERFORM F9200_GOOD_RECEIPT TABLES I_CAT3
                                           USING  V_BWART V_WERKS
                                                  V_LGORT V_MATNR
                                                  V_QTY   V_KOSTL
                                                  V_POSTCALQTY
                                                  V UOM.
              ELSE.
```

```
* Move data for print test mode
                MOVE : I_CAT3 TO I_TEST.
                MOVE : V POSTCALQTY TO I TEST-AMT POST .

                APPEND I_TEST.
                CLEAR  I_TEST.
              ENDIF.  " End check test mode

            ENDIF.  " End of Check Exchange Rate V RATE is initial

          ENDIF.  " End of check movement type 901,521

        ENDIF.  " End of check steam material for categories 3

      ENDLOOP.  " End loop of I CAT3

    ENDIF.  " End IF NOT I CAT3[] IS INITIAL.

  ENDIF.  " End IF of check box -> RB CAT1, RB CAT2 and RB CAT3

ENDFORM.                           " F4100 POSTING CATEGORY

*********************************************************************
       FORM F4110 PORTION WITH RATIO

       Description: This form is use for ratio calculation, that
                    formula will get ratio from ZCPCSCOST2 for
                    posting for category 2.
*********************************************************************
FORM F4110 PORTION WITH_ RATIO.

  LOOP AT I_POST CAT2.

    CLEAR : V_RATIO_1
            V QTY_LINE    ,
            V_CAL RATIO 1,
            V POST        .

    READ TABLE I_GET_RATIO WITH KEY KOSTL = I_POST_CAT2-KOSTL
                                    MATNR = I POST CAT2-MATNR.

    IF SY-SUBRC = 0.

      MOVE : I_GET_RATIO-RATIO TO V_RATIO_1    ,
             I POST CAT2-ERFMG TO V QTY LINE1 .

      MOVE : I_POST_CAT2-BWART  TO V_BWART
             I_POST_CAT2-WERKS  TO V_WERKS
             I_POST_CAT2-LGORT  TO V_LGORT
             I_POST_CAT2-MATNR  TO V_MATNR
             I_POST_CAT2-ERFMG  TO V_QTY
             I_POST_CAT2-KOSTL  TO V_KOSTL
             I POST CAT2-UOM     TO V UOM

* Ratio of Material 1 = (Quantity material 1 * Ratio material 1) /
                       ((Quantity material 1 * Ratio material 1) +
                        (Quantity material 2 * Ratio material 2) +

                        (Quantity material 2 * Ratio material 2)).
      V_CAL_RATIO_1 = ( V_QTY_LINE1 * V_RATIO_1 ) / V_TOTAL_RATIO.

* Posted amount in SGD = [Difference from Cost center USD] * Ratio *
                         exchange rate(TCURR).

      V POST LINE1 = I VALUE-WKGBTR * V CAL RATIO 1 * V RATE.

      MOVE : V POST LINE1 TO V POSTCALQTY.

* Check test mode before posting
      IF P TEST = ''.

* Posting BDC for Good Receipt (MB1C).
        PERFORM F9200_GOOD_RECEIPT TABLES I_POST_CAT2
                                   USING  V_BWART V_WERKS
                                          V_LGORT V_MATNR
                                          V_QTY   V_KOSTL
                                          V POSTCALQTY
                                          V_UOM.
      ELSE.
        MOVE-CORRESPONDING I POST CAT2 TO I TEST.

        MOVE : V POSTCALQTY TO I TEST-AMT POST .

        APPEND I_TEST.
        CLEAR  I_TEST.
      ENDIF.  " End check test mode
    ENDIF.  " End check sub-rc of I_GET RATIO

  ENDLOOP.  " End loop of I_POST_CAT2

  REFRESH : I_POST_CAT2.
  CLEAR   : I_POST_CAT2  ,
            V_TOTAL_RATIO.

ENDFORM.                     " F4110_PORTION_WITH_RATIO

**********************************************************************
       FORM F4120 POST WITHOUT RATIO

•      Description: This form is use for non ratio calculation, that*
                    formula for posting is Amount to be posted =
                    [Difference from Cost Center USD] * Exchange
                    rate.
                    This perform will posting for category 2.
**********************************************************************
FORM F4120_POST_WITHOUT_RATIO.

  IF I CATEGORY2-BWART = C 901.

    CLEAR : V_BWART ,
            V_WERKS ,
            V_LGORT ,
            V_MATNR ,
            V_QTY   ,
            V_KOSTL ,
            V UOM

    MOVE : I_CATEGORY2-BWART TO V_BWART,
           I CATEGORY2-WERKS TO V WERKS,
      I_CATEGORY2-LGORT TO V_LGORT,
      I_CATEGORY2-MATNR TO V_MATNR,
      I_CATEGORY2-ERFMG TO V_QTY
      I_CATEGORY2-KOSTL TO V_KOSTL,
      I CATEGORY2-UOM  TO V UOM  .

* Check test mode before posting
  IF P TEST = ''.

* Posting BDC for Good Issues (MB1A).
     PERFORM F9100_GOOD_ISSUES TABLES I_CATEGORY2
                               USING  V_BWART V_WERKS V_LGORT
                                      V_MATNR V_QTY   V_KOSTL
                                      V_UOM.
  ELSE.
    MOVE-CORRESPONDING I CATEGORY2 TO I TEST.

    I TEST-AMT POST = 0.

    APPEND I_TEST.
    CLEAR  I_TEST.
  ENDIF.

 ELSEIF I CATEGORY2-BWART = C 521.

  CLEAR : V_BWART ,
          V_WERKS ,
          V_LGORT ,
          V_MATNR ,
          V_PRICE ,
          V_ERFMG ,
          V_QTY
          V_KOSTL ,
          V_POSTCALQTY,
          V_POST_AMOUNT,
          V_UOM    .

* Check Exchange Rate
  IF V RATE IS INITIAL.

    MOVE-CORRESPONDING I CATEGORY2 TO I ERROR.

    I ERROR-AMT POST = 0.

* Exchange Rate Not Found!
    MOVE : 'Exchange Rate Not Found!'(046) TO I_ERROR-REMARK.

    APPEND I_ERROR.
    CLEAR  I_ERROR.
  ELSE.

* Found Exchange rate

    MOVE : I_CATEGORY2-ERFMG TO V_ERFMG,
           I CATEGORY2-PRICE TO V_PRICE.

    READ TABLE I VALUE WITH KEY OBJNR+12(4) = I CATEGORY2-KOSTL.

    IF SY-SUBRC = 0.
```

```
* Posted amount in SGD = (Difference from Cost center USD] *
                        exchange rate(TCURR)
        V POST AMOUNT = I VALUE-WKGBTR * V RATE.

        MOVE : V POST AMOUNT TO V POSTCALQTY.

        MOVE : I_CATEGORY2-BWART  TO V_BWART
               I CATEGORY2-WERKS  TO V_WERKS
               I_CATEGORY2-LGORT  TO V_LGORT
               I_CATEGORY2-MATNR  TO V_MATNR
               I_CATEGORY2-ERFMG  TO V_QTY
               I_CATEGORY2-KOSTL  TO V_KOSTL
               I CATEGORY2-UOM     TO V UOM

* Check test mode before posting
        IF P_TEST = ''.

* Posting BDC for Good Receipt (MB1C).
          PERFORM F9200_GOOD_RECEIPT TABLES I_CATEGORY2
                                     USING  V_BWART V_WERKS
                                            V_LGORT V_MATNR
                                            V_QTY   V_KOSTL
                                            V_POSTCALQTY
                                            V UOM.
        ELSE.
          MOVE-CORRESPONDING I CATEGORY2 TO I TEST.

          MOVE : V POSTCALQTY TO I TEST-AMT POST .

          APPEND I_TEST.
          CLEAR I TEST.
        ENDIF.  " End check test mode
      ENDIF.  " End READ I_VALUE
    ENDIF.  " End of Check Exchange Rate V_RATE is initial
  ENDIF.  " End of check movement type 901,521

ENDFORM.                        " F4120 POST WITHOUT RATIO

**********************************************************************
•      FORM F5000_PRINT_REPORT
*                                                                    *
•      Description: This form will generate report both of success  *
                    and error report for each categories type.
**********************************************************************
FORM F5000 PRINT REPORT.

* Check test mode before posting
  IF P_TEST = ''.

* Print Success Report
    PERFORM F5100_WRITE_REPORT TABLES I_SUCCESS
                               USING C SUCCESS HEAD.

* Print Error Report
    PERFORM F5100_WRITE_REPORT TABLES I_ERROR
                               USING C ERROR HEAD.

  ELSE.

    IF NOT I ERROR[] IS INITIAL.
```

```
* Print Error Report
      PERFORM F5100_WRITE_REPORT TABLES I_ERROR
                                USING C ERROR HEAD.

    ENDIF.

    IF NOT I TEST[] IS INITIAL.

* Print Test Report
      PERFORM F5100_WRITE_REPORT TABLES I_TEST
                                USING C TEST HEAD.
    ENDIF.

  ENDIF.

ENDFORM.                          " F5000_PRINT_REPORT

*********************************************************************
       FORM F5100_WRITE_REPORT
*                                                                   *
       Description: This form will generate success for all
                    categories order by categories.
*********************************************************************
FORM F5100 WRITE REPORT TABLES I_PRINT STRUCTURE I SUCCESS
                        USING V HEAD.


  FORMAT COLOR COL HEADING INTENSIFIED ON.

  WRITE : /1     V HEAD,
           135

  FORMAT RESET.

* Print report
  LOOP AT I_PRINT.
    WRITE : /1   I_PRINT-CATG                        " CAT NO
             10  I_PRINT-SEQUE                       " SEQUENCE
             20  I_PRINT-MATNR                       " MAT CODE
             40  I_PRINT-BWART                       " MOVEMENT TYPE
             50  I_PRINT-KOSTL                       " COST CENTER
             63  I_PRINT-WERKS                       " PLANT
             70  I PRINT-LGORT                       " STORAGE
             79  I_PRINT-ERFMG                       " QUANTITY
             92  I_PRINT-UOM                         " UOM
             97  I_PRINT-PRICE                       " UNIT PRICE
             115 I PRINT-AMT POST CURRENCY C_SGD.    " AMT POST

    WRITE : /1   I PRINT-REMARK                      " REMARK

  ENDLOOP.

ENDFORM.                          " F5100 WRITE REPORT

*********************************************************************
       FORM F9100_GOOD_ISSUES
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
       Description: This form will posting good issue via
                    transaction code MB1A with movement type = 961.
```

```
*****************************************************************************
FORM F9100_GOOD_ISSUES TABLES I_GOOD_ISSUES STRUCTURE I_TMPDATA
                       USING  V_BWART V_WERKS V_LGORT V_MATNR
                              V QTY    V KOSTL V UOM.

  REFRESH : I_RET_MSG,
            I BDCDATA.

  CLEAR : I_RET_MSG  ,
          I BDCDATA  .

  PERFORM BDC_DYNPRO      USING 'SAPMM07M'       '0400'.
  PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '/00'.
  PERFORM BDC FIELD       USING 'BDC CURSOR'     'RM07M-LGORT'.

  PERFORM BDC_FIELD       USING 'MKPF-BLDAT'     V_DOCDATE.
  PERFORM BDC_FIELD       USING 'MKPF-BUDAT'     V_POSTDATE.
  PERFORM BDC_FIELD       USING 'MKPF-BKTXT'     V_PIS_TEXT.
  PERFORM BDC_FIELD       USING 'RM07M-BWARTWA'  V_BWART.
  PERFORM BDC_FIELD       USING 'RM07M-WERKS'    V_WERKS.
  PERFORM BDC_FIELD       USING 'RM07M-LGORT'    V_LGORT.
  PERFORM BDC_FIELD       USING 'XFULL'          'X'.
  PERFORM BDC FIELD       USING 'RM07M-WVERS2'   'X'.

  PERFORM BDC_DYNPRO      USING 'SAPMM07M'       '0421'.
  PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '/00 .
  PERFORM BDC FIELD       USING 'BDC CURSOR'     'MSEG-ERFME(01)'.

  PERFORM BDC_FIELD       USING 'MSEG-MATNR(01)' V_MATNR.
  PERFORM BDC_FIELD       USING 'MSEG-ERFMG(01)' V_QTY.
  PERFORM BDC_FIELD       USING  MSEG-ERFME(01)' V_UOM.
  PERFORM BDC_FIELD       USING 'BDC_SUBSCR'     'SAPMM07M'.
  PERFORM BDC FIELD       USING 'BDC SUBSCR'     'SAPLKACB'.

  PERFORM BDC_DYNPRO      USING 'SAPLKACB'       '0002'.
  PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '=ENTE'.
  PERFORM BDC FIELD       USING 'BDC CURSOR'     'COBL-KOSTL'.

  PERFORM BDC_FIELD       USING 'COBL-KOSTL'     V_KOSTL.
  PERFORM BDC FIELD       USING 'BDC SUBSCR'     'SAPLKACB'.

  PERFORM BDC_DYNPRO      USING 'SAPLKACB'       '0002'.
  PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '=ENTE'.
  PERFORM BDC FIELD       USING 'BDC CURSOR'     'COBL-KOSTL'.

  PERFORM BDC_FIELD       USING 'COBL-KOSTL'     V_KOSTL.
  PERFORM BDC FIELD       USING 'BDC SUBSCR'     'SAPLKACB'.

  PERFORM BDC_DYNPRO      USING 'SAPMM07M'       '0421'.
  PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '=BU'.
  PERFORM BDC FIELD       USING 'BDC CURSOR'     'MSEG-ERFMG(01) .

  PERFORM BDC_FIELD       USING 'BDC_SUBSCR'     'SAPMM07M'.
  PERFORM BDC_FIELD       USING 'BDC_SUBSCR'     'SAPLKACB'.
  PERFORM BDC_FIELD       USING 'DKACB-FMORE'    'X'.

  PERFORM BDC_DYNPRO      USING 'SAPLKACB'       '0002'.
  PERFORM BDC FIELD       USING 'BDC OKCODE'     '=ENTE'.

  PERFORM BDC_FIELD       USING 'BDC_CURSOR'     'COBL-KOSTL'.
  PERFORM BDC FIELD       USING 'BDC SUBSCR'      'SAPLKACH'.

* BDC Process for transaction 'MB1A'
  CALL TRANSACTION C_MB1A USING    I_BDCDATA
                          MODE     V_MODE
                          UPDATE   C_UPDATE
                          MESSAGES INTO I RET MSG.

* For display success message only incase of success.
  READ TABLE I_RET_MSG WITH KEY MSGTYP = C_S
                                MSGID  = C_M7
                                MSGNR  = C 060.

  IF SY-SUBRC = 0.
* Success posting

    CLEAR : V REMARK.

* Call function for translate message
    CALL FUNCTION 'FORMAT_ MESSAGE'
         EXPORTING
             ID   = I_RET_MSG-MSGID
             LANG = SY-LANGU
             NO   = I_RET_MSG-MSGNR
             V1   = I_RET_MSG-MSGV1
             V2   = I_RET_MSG-MSGV2
             V3   = I_RET_MSG-MSGV3
             V4   = I_RET_MSG-MSGV4
         IMPORTING
             MSG  = V REMARK.

    MOVE : I_GOOD_ISSUES-CATG  TO  I_SUCCESS-CATG ,
           I_GOOD_ISSUES-SEQUE TO  I_SUCCESS-SEQUE,
           I_GOOD_ISSUES-MATNR TO  I_SUCCESS-MATNR,
           I_GOOD_ISSUES-BWART TO  I_SUCCESS-BWART,
           I_GOOD_ISSUES-KOSTL TO  I_SUCCESS-KOSTL,
           I_GOOD_ISSUES-WERKS TO  I_SUCCESS-WERKS,
           I_GOOD_ISSUES-LGORT TO  I_SUCCESS-LGORT,
           I GOOD_ISSUES-ERFMG TO  I_SUCCESS-ERFMG,
           I_GOOD_ISSUES-PRICE TO  I_SUCCESS-PRICE,
           I GOOD ISSUES-UOM   TO  I_SUCCESS-UOM   .

    CONCATENATE I_RET_MSG-MSGV1 SY-DATUM+0(4) V_REMARK
           INTO I_SUCCESS-REMARK
           SEPARATED BY SPACE.

    APPEND I_SUCCESS.
    CLEAR  I_SUCCESS.

  ELSE.
    LOOP AT I_RET_MSG.
* Error posting

      CLEAR : V REMARK.

* Call function for translate message
      CALL FUNCTION 'FORMAT_ MESSAGE'
           EXPORTING
               ID   = I_RET_MSG-MSGID
               LANG = SY-LANGU
               NO    = I_RET_MSG-MSGNR
               V1    = I_RET_MSG-MSGV1
               V2    = I_RET_MSG-MSGV2
               V3    = I_RET_MSG-MSGV3
               V4    = I_RET_MSG-MSGV4
           IMPORTING
               MSG   = V REMARK.

      MOVE : I_GOOD_ISSUES-CATG   TO  I_ERROR-CATG ,
             I_GOOD_ISSUES-SEQUE  TO  I_ERROR-SEQUE,
             I_GOOD_ISSUES-MATNR  TO  I_ERROR-MATNR,
             I_GOOD_ISSUES-BWART  TO  I_ERROR-BWART,
             I_GOOD_ISSUES-KOSTL  TO  I_ERROR-KOSTL,
             I_GOOD_ISSUES-WERKS  TO  I_ERROR-WERKS,
             I_GOOD_ISSUES-LGORT  TO  I_ERROR-LGORT,
             I_GOOD_ISSUES-ERFMG  TO  I_ERROR-ERFMG,
             I_GOOD_ISSUES-PRICE  TO  I_ERROR-PRICE,
             I GOOD ISSUES-UOM    TO  I ERROR-UOM.

      CONCATENATE I_RET_MSG-MSGID I_RET_MSG-MSGTYP I_RET_MSG-MSGNR
                  C_COLON V_REMARK
                  INTO I_ERROR-REMARK
                  SEPARATED BY SPACE.

      APPEND I_ERROR.
      CLEAR  I_ERROR.

    ENDLOOP.
  ENDIF.

ENDFORM.                              " F9100 GOOD ISSUES

**********************************************************************
       FORM F9200_GOOD_RECEIPT
*--------------------------------------------------------------------*
       Description: This form will posting good receipt via
                    transaction code MB1C with movement type = 521. *
**********************************************************************
FORM F9200_GOOD_RECEIPT TABLES I_GOOD_RECEIPT STRUCTURE I_TMPDATA
                        USING  V_BWART V_WERKS V_LGORT V_MATNR
                               V_QTY   V_KOSTL V_POSTCALQTY V_UOM.
* Posting incase of posting quantity <> 0.
  IF V_POSTCALQTY <> 0.
    REFRESH : I_RET_MSG,
              I BDCDATA.

    CLEAR : I_RET_MSG   ,
            I BDCDATA   .

    PERFORM BDC_DYNPRO      USING 'SAPMM07M'       '0400'.
    PERFORM BDC FIELD       USING 'BDC_OKCODE'     '/00'.
    PERFORM BDC FIELD       USING 'BDC CURSOR'     'RM07M-LGORT'.

    PERFORM BDC_FIELD       USING 'MKPF-BLDAT'     V_DOCDATE.
    PERFORM BDC_FIELD       USING 'MKPF-BUDAT'     V_POSTDATE.
    PERFORM BDC FIELD       USING 'MKPF-BKTXT'     V PIS TEXT.

    PERFORM BDC_FIELD       USING 'RM07M-BWARTWA'  V_BWART.
    PERFORM BDC_FIELD       USING 'RM07M-WERKS'    V_WERKS.
    PERFORM BDC FIELD       USING 'RM07M-LGORT'    V LGORT.
    PERFORM BDC_FIELD       USING 'XFULL'
    PERFORM BDC_FIELD       USING 'RMO7M-WVERS2 '' X ' .

    PERFORM BDC_DYNPRO      USING 'SAPMMO7M'       '0421'.
    PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '/00'.
    PERFORM BDC FIELD       USING 'BDCCURSOR'      'MSEG-MATNR(01)'.

    PERFORM BDC_FIELD       USING 'MSEG-MATNR(01)' V_MATNR.
    PERFORM BDC_FIELD       USING 'MSEG-ERFMG(01)' V_QTY.
    PERFORM BDC_FIELD       USING 'MSEG-ERFME(01)' V_UOM.
    PERFORM BDC_FIELD       USING 'BDC_SUBSCR'     'SAPMM07M'.
    PERFORM BDC_FIELD       USING 'BDC_SUBSCR'     'SAPLKACB'.
    PERFORM BDC FIELD       USING 'COBL-KOSTL'     V KOSTL.

    PERFORM BDC_DYNPRO      USING 'SAPMM07M'       '0421'.
    PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '=KPA'.
    PERFORM BDC_FIELD       USING 'BDC CURSOR'     'MSEG-ERFMG(01)'.

    PERFORM BDC_FIELD       USING 'BDC_SUBSCR'     'SAPMM07M'.
    PERFORM BDC FIELD       USING 'BDC SUBSCR'     'SAPLKACB'.

    PERFORM BDC_DYNPRO      USING 'SAPMM07M'       '0410'.
    PERFORM BDC_FIELD       USING 'BDC_OKCODE'     '=BU'.
    PERFORM BDC FIELD       USING 'BDC_CURSOR'     'MSEG-EXBWR'.

    PERFORM BDC_FIELD       USING 'MSEG-ERFMG'    V_QTY.
    PERFORM BDC_FIELD       USING 'MSEG-EXBWR'    V_POSTCALQTY.
    PERFORM BDC_FIELD       USING 'BDC_SUBSCR'    'SAPMM07M'.
    PERFORM BDC_FIELD       USING 'BDC_SUBSCR'    'SAPLKACB'.
    PERFORM BDC FIELD       USING 'COBL-KOSTL'    V_KOSTL.

* BDC Process for transaction 'MB1C'
    CALL TRANSACTION C_MB1C USING    I_BDCDATA
                            MODE     V_MODE
                            UPDATE   C_UPDATE
                            MESSAGES INTO I RET MSG.

  READ TABLE I_RET_MSG WITH KEY MSGTYP = C_S
                                MSGID  = C_M7
                                MSGNR  = C_060.

  IF SY-SUBRC = O.
* Success posting
    CLEAR : V REMARK.

* Call function for translate message
    CALL FUNCTION 'FORMAT_ MESSAGE'
         EXPORTING
             ID   = I_RET_MSG-MSGID
             LANG = SY-LANGU
             NO   = I_RET_MSG-MSGNR
             V1   = I_RET_MSG-MSGV1
             V2   = I_RET_MSG-MSGV2
             V3   = I_RET_MSG-MSGV3
             V4   = I_RET_MSG-MSGV4
         IMPORTING
             MSG  = V_REMARK.

    MOVE : V POSTCALQTY TO I SUCCESS-AMT POST .
    MOVE : I_GOOD_RECEIPT-CATG   TO  I_SUCCESS-CATG ,
           I_GOOD_RECEIPT-SEQUE  TO  I_SUCCESS-SEQUE,
           I_GOOD_RECEIPT-MATNR  TO  I_SUCCESS-MATNR,
           I_GOOD_RECEIPT-BWART  TO  I_SUCCESS-BWART,
           I_GOOD_RECEIPT-KOSTL  TO  I_SUCCESS-KOSTL,
           I_GOOD_RECEIPT-WERKS  TO  I_SUCCESS-WERKS,
           I_GOOD_RECEIPT-LGORT  TO  I_SUCCESS-LGORT,
           I_GOOD_RECEIPT-ERFMG  TO  I_SUCCESS-ERFMG,
           I_GOOD_RECEIPT-PRICE  TO  I_SUCCESS-PRICE,
           I GOOD RECEIPT-UOM    TO  I SUCCESS-UOM   .

    CONCATENATE I_RET_MSG-MSGV1 SY-DATUM+0(4) V_REMARK
           INTO I_SUCCESS-REMARK
           SEPARATED BY SPACE.

    APPEND I_SUCCESS.
    CLEAR  I_SUCCESS.

  ELSE.

* Error posting
    LOOP AT I_RET_MSG.
      CLEAR : V REMARK.

* Call function for translate message
      CALL FUNCTION 'FORMAT_MESSAGE'
           EXPORTING
               ID   = I_RET_MSG-MSGID
               LANG = SY-LANGU
               NO   = I_RET_MSG-MSGNR
               V1   = I_RET_MSG-MSGV1
               V2   = I_RET MSG-MSGV2
               V3   = I_RET_MSG-MSGV3
               V4   = I_RET_MSG-MSGV4
           IMPORTING
               MSG  = V REMARK.

      MOVE : V POSTCALQTY TO I ERROR-AMT POST .

      MOVE : I_GOOD_RECEIPT-CATG  TO  I ERROR-CATG ,
             I_GOOD_RECEIPT-SEQUE TO  I ERROR-SEQUE,
             I_GOOD_RECEIPT-MATNR TO  I ERROR-MATNR,
             I_GOOD_RECEIPT-BWART TO  I_ERROR-BWART,
             I_GOOD_RECEIPT-KOSTL TO  I ERROR-KOSTL,
             I_GOOD_RECEIPT-WERKS TO  I ERROR-WERKS,
             I_GOOD_RECEIPT-LGORT TO  I_ERROR-LGORT,
             I_GOOD_RECEIPT-ERFMG TO  I_ERROR-ERFMG,
             I_GOOD_RECEIPT-PRICE TO  I_ERROR-PRICE,
             I_GOOD_RECEIPT-UOM    TO  I ERROR-UOM.

      CONCATENATE I RET_MSG-MSGID I_RET MSG-MSGTYP I RET MSG-MSGNR
                  C COLON V_REMAR
                  INTO I ERROR  EMARK
                  SEPARATED BY SPACE.

      APPEND I_ERROR.
      CLEAR  I_ERROR.
    ENDLOOP.
  ENDIF.
  ELSE.

* Move data to error report because posting quantity = 0.
    MOVE : V POSTCALQTY TO I ERROR-AMT POST .

    MOVE : I GOOD RECEIPT-CATG   TO  I ERROR-CATG ,

           I GOOD RECEIPT-SEQUE  TO  I_ERROR-SEQUE,
           I GOOD RECEIPT-MATNR  TO  I_ERROR-MATNR,
           I GOOD RECEIPT-BWART  TO  I_ERROR-BWART,
           I_GOOD_RECEIPT-KOSTL  TO  I_ERROR-KOSTL,
           I_GOOD_RECEIPT-WERKS  TO  I_ERROR-WERKS,
           I_GOOD_RECEIPT-LGORT  TO  I_ERROR-LGORT,
           I_GOOD_RECEIPT-ERFMG  TO  I_ERROR-ERFMG,
           I_GOOD_RECEIPT-PRICE  TO  I_ERROR-PRICE,
           I_GOOD_RECEIPT-UOM    TO  I ERROR-UOM   .

* Posting quantity is ZERO!
    MOVE : 'Posting quantity is ZERO!'(071)
      TO I ERROR-REMARK.

    APPEND I_ERROR.
    CLEAR  I_ERROR.

  ENDIF.  " ENDIF V POSTCALQTY <> 0.

ENDFORM.                          " F9200 GOOD RECEIPT

**********************************************************************
       FORM F9300 TOP OF PAGE                                        *
                                                                     *
       Description: This form will print top of page by using        *
                    standard header ZREPHEAD                         *
**********************************************************************
FORM F9300_TOP OF PAGE.

* Call standard report heading
  SUMMARY.
  FORMAT COLOR COL HEADING.

* Write header report
  PERFORM HEADER(ZREPHEAD) USING
                            V_REPID         " report-id
                            SY-LINSZ.       " line-size

* Write header
  WRITE : /1    'CAT NO'(011),
           10   'SEQUENCE'(012),
           20   'MAT CODE'(013),
           40   'MVT TYPE'(014),
           50   'COST CENTER'(015),
           63   'PLANT'(016),
           70   'STORAGE'(017),
           79   'QUANTITY'(018),
           92   'UOM'(084),
           97   'UNIT PRICE'(019),
           115  'POST AMOUNT'(020),
           135  ' '

  WRITE : /1    'REMARK'(021),
          135 ' '

ENDFORM.                            " F9300 TOP OF PAGE

**********************************************************************
       FORM BDC_DYNPRO
*--------------------------------------------------------------------*
       Description: This form will posting BDC DYNPRO.
**********************************************************************
FORM BDC_DYNPRO USING PROGRAM DYNPRO.

  CLEAR : I BDCDATA.

  I_BDCDATA-PROGRAM  = PROGRAM.
  I_BDCDATA-DYNPRO   = DYNPRO.
  I BDCDATA-DYNBEGIN = C X.

  APPEND I_BDCDATA.

ENDFORM.                            " BDC DYNPRO

**********************************************************************
       FORM BDC FIELD

       Description: This form will posting BDC FIELD.
**********************************************************************
FORM BDC_FIELD USING FNAM FVAL.

  CLEAR : I BDCDATA.

  I_BDCDATA-FNAM = FNAM.
  I BDCDATA-FVAL = FVAL.

  APPEND I_BDCDATA.

ENDFORM.                            " BDC FIELD
```

APPENDIX B: Source Code of Outbound Customized Application Integration Program for workstation.

```
REPORT ZOUTBINT LINE-SIZE 120
                LINE-COUNT 65
                NO STANDARD PAGE HEADING.

*********************************************************************
*  Program            : ZOUTBINT
*  Author             : Supaporn Wongwithit
*  Created            : 03/04/2002
•  Transport No.
*
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
•  Description        : Interface program between SAP and another
                        system.
                        This program will transfer data from SAP to
                        another system. This program will retrieve
                        data from SAP database in last half and hour.
                        And populate/reformat data assign batch no,
                        store data into Backup database.
                        After that generate Outbound file send to
                        another system.
*
•  Input
*
•  Output             : Text file and report for success and unsuccess
                        records.
*
*  Called from
*
*  Called to
*
*  Includes
*
*  Function Modules   :
*
•  Logical Database
*
•  High Level Design :
*
* - - - - - - - - - - ______________________________

*********************************************************************
•     TABLES                                                        *
*********************************************************************
TABLES: ZOUTBOUND1 ,          " Outbound format 1
        ZOUTBOUND2 ,          " Outbound format 2
        ZOUTBOUND3 ,          " Outbound format 3
        ZBATCH_CONTROL_O.     " Outbound Batch Control

*********************************************************************
•     DATA
*********************************************************************
  - - - - - Constants  - - - - - - - - - - - - - - - - - - - - - - -
CONSTANTS: C_DEFAULT_PATH(50) TYPE C VALUE 'C:\TEMP'    , " Path unix
           C_FILENAME(50)     TYPE C VALUE '\TPCTOPSA' , " File name
           C_DAT(3)          TYPE C VALUE 'DAT'         " DAT file
           C_TXT(4)          TYPE C VALUE '.TXT'        " Txt file
           C_102(3)          TYPE C VALUE '102'         " Trans type
           C_107(3)          TYPE C VALUE '107'         " Trans type
           C_108(3)          TYPE C VALUE '108'         " Trans type
           C_109(3)          TYPE C VALUE '109'         " Trans type
           C_111(3)          TYPE C VALUE '111'         " Trans type
           C_112(3)          TYPE C VALUE '112'         " Trans type
           C_113 (3)         TYPE C VALUE '113'         " Trans type
           C_114(3)          TYPE C VALUE '114'         " Trans type
           C_116(3)          TYPE C VALUE '116'         " Trans type
           C_01500(5)        TYPE C VALUE '01500'       " Rec size
           C_COLON(1)        TYPE C VALUE ' -
           C 01              LIKE INRI-NRRANGENR
                                    VALUE '01'        , " Interval#
           C_ZOUTBATCHC      LIKE INRI-OBJECT
                                    VALUE 'ZOUTBATCHC'    Obj#

  - - - -  Working areas  - - - - - - - - - - - - - - - - - - - - -

  - - - -  Internal table  - - - - - - - - - - - - - - - - - - - - -
TYPES: BEGIN OF TYP_HEADER,
         HEADER(100)    TYPE C ,           " Header
       END OF TYP_HEADER.

DATA : I_TMPDATA1      LIKE ZOUTBOUND1 OCCURS 0 WITH HEADER LINE.
DATA : I_TMPDATA2      LIKE ZOUTBOUND2   OCCURS 0  WITH HEADER LINE.
DATA : I_TMPDATA3      LIKE ZOUTBOUND3   OCCURS 0  WITH HEADER LINE.
DATA : I HEADER        TYPE TYP HEADER   OCCURS 0  WITH HEADER LINE.

  - - - - - Variable  - - - - - - - - - - - - - - - - - - - - - - -
DATA : V_REPID         LIKE SY-REPID,    Report-id
       V_LINES1        TYPE N ,          Lines from format file 1
       V_LINES2        TYPE N ,          Lines from format file 2
       V_LINES3        TYPE              Lines from format file 3
       V_HSUMLINES(6) TYPE N ,           Sum Lines include header
       V_SUMLINES(6)   TYPE N ,          Sum Lines from format file
       V_CSUMLINES(6) TYPE C             Char for Sum lines of file
       V_REP1          TYPE              Flag chk for report type 1
       V_REP3          TYPE N            Flag chk for report type 3
       V_DATE          LIKE SY-DATUM , " System Date
       V_TIME          LIKE SY-UZEIT , " System Time
       V_BATCH_NO      LIKE ZBATCH_CONTROL_O-BATCH_NO , " Batch#
       V_TMPBATCH_NO(6)TYPE C          , " Batch # For temp
       V DEFAULT NAME LIKE RLGRAP-FILENAME  . " File Name

*********************************************************************
       SELECTION SCREEN
*********************************************************************
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.

PARAMETERS: P_TEST AS CHECKBOX default 'X'.      " Test mode chk box

SELECTION-SCREEN END OF BLOCK B1.

*********************************************************************
       INITIALIZATION                                                *
*********************************************************************
INITIALIZATION.
```

```
MOVE SY-REPID TO V REPID.

**********************************************************************
       TOP  OF  PAGE
**********************************************************************
TOP-OF-PAGE.

* -- Call standard report heading
  SUMMARY.
  FORMAT COLOR COL_HEADING.

* -- Write header report
  PERFORM   HEADER(ZREPHEAD) USING
                               V_REPID                 " report-id
                               SY-LINSZ.               " line-size

  WRITE : /1    'ITEM NO'(010),
           15   'INSTUCTION DATE'(011),
           35   'INSTUCTION TIME'(012),
           55   'GOODS CODE'(013).

**********************************************************************
       BEGIN    SELECTION
**********************************************************************
START-OF-SELECTION.

*-- For test only Prototype Phase
  IF P TEST = 'X'.

*-- Get data from SAP table
    PERFORM 01000 GET DATA.

*-- Generate Batch number
    PERFORM 02000_GEN_BATCH_NUMBER.

  - Check internal table before process
    IF V_SUMLINES > 0.

  - Interface data : Send data from SAP to another system
      PERFORM 03000_DATA_INTERFACE.

*-- Print Report
      PERFORM 04000_PRINT_REPORT.

    ELSE.
      MESSAGE S000(ZY) WITH
      'No data Selected!!!'(020).
    ENDIF.

*-- End of P_TEST = 'X'.
  ENDIF.

END-OF-SELECTION.

**********************************************************************
•      END  OF  SELECTION
**********************************************************************

**********************************************************************
       BEGIN    FORMS
```

```
*********************************************************************

*********************************************************************
       FORM 01000_GET_DATA
* ------------------------------------------------------------------
       Description: Select data from customized table that have data*
                    from user exit and this data will be transfer to*
                    text file and send to another system by using   *
                    internal table I DATA.
*********************************************************************
FORM 01000_GET_DATA.

*-- Clear internal table
  REFRESH : I_TMPDATA1,
            I TMPDATA3.

  CLEAR   : I_TMPDATA1,
            I_TMPDATA3,
            V_LINES1  ,
            V_LINES2  ,
            V_LINES3
            V_SUMLINES,
            V_CSUMLINES,
            V HSUMLINES.

*-- Select data from SAP table and insert into internal table before
*-- send it out to another system.

  SELECT *
    INTO TABLE I_TMPDATA1
    FROM ZOUTBOUND1
   WHERE BATCH NO = ''

  DESCRIBE TABLE I_TMPDATA1 LINES V LINES1.

  SELECT *
    INTO TABLE I_TMPDATA3
    FROM ZOUTBOUND3
   WHERE BATCH NO = ''

  DESCRIBE TABLE I_TMPDATA3 LINES V_LINES3.

*-- Total Lines of data
  V SUMLINES = V LINES1 + V LINES2 + V LINES3.

*-- Total Lines include header line.
  V HSUMLINES = V SUMLINES + 1.

*-- should add some where clause for get only last 1/2 hr. for send
it
*-- out.

ENDFORM.                          " 01000_GET_DATA

*********************************************************************
       FORM 02000 GEN BATCH NUMBER

       Description: This Form will generate batch number and update *
                    batch number into ZOUTBOUNDX internal table for *
                    each file format
```

```
**********************************************************************
FORM 02000 GEN BATCH NUMBER.

  REFRESH : I HEADER  .

  CLEAR   : I_HEADER
            V_BATCH_NO,
            V_DATE
            V_TIME

  CALL FUNCTION 'NUMBER_GET_NEXT
    EXPORTING
      NR_RANGE_NR                  = C_01
      OBJECT                       = C_ZOUTBATCHC
    IMPORTING
      NUMBER                       = V_BATCH_NO
    EXCEPTIONS
      INTERVAL_NOT_FOUND           = 1
      NUMBER_RANGE_NOT_INTERN      = 2
      OTHERS                       = 8.

    IF SY-SUBRC <> 0.
      MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
      WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
    ELSE.
      MOVE : SY-DATUM TO V_DATE,
             SY-UZEIT TO V_TIME.

*-- Update batch number into ZBATCH_CONTROL 0.
      ZBATCH CONTROL 0-BATCH NO = V  BATCH

      INSERT ZBATCH_CONTROL_O.

      V TMPBATCH NO = V BATCH NO+4(6).

*-- Generate File name
      CONCATENATE C_DEFAULT PATH C FILENAME V TMPBATCH_NO C_TXT
             INTO V_DEFAULT

*-- Total lines include header line
      UNPACK V HSUMLINES TO V CSUMLINES.

  - File header example : 2002021811:20:0100005201500016962
      CONCATENATE V_DATE V_TIME+0(2) C_COLON
                  V_TIME+2(2) C_COLON V_TIME+4(2) V_CSUMLINES
                  C_01500 V_TMPBATCH_NO
             INTO I_HEADER-HEADER.

      APPEND I_HEADER.
    ENDIF.

    CALL FUNCTION 'WS_DOWNLOAD'
      EXPORTING
        FILENAME  =  V_DEFAULT_NAME
        FILETYPE  =  C_DAT
      TABLES
        DATA TAB  = I HEADER.

    IF SY-SUBRC <> 0.
      MESSAGE E000(ZY) WITH 'Unable to download header'(002).
    ENDIF.

ENDFORM.                          " 02000 GEN BATCH NUMBER

*********************************************************************
       FORM 03000_DATA_INTERFACE

       Description: This Form will send data out from SAP to another*
                    System
*********************************************************************
FORM 03000_DATA_INTERFACE.

  IF SY-SUBRC <> 0.
     EXIT.
  ENDIF.

*-- Check for download file format 1
  IF V_LINES1 > 0.

*-- Flag check download success = 0. unsuccess = 1.
    V REP1 = 0.

    CALL FUNCTION 'WS_DOWNLOAD'
      EXPORTING
        FILENAME  =  V_DEFAULT_NAME
        FILETYPE  =  C_DAT
      TABLES
        DATA TAB  =  I TMPDATA1.

    IF SY-SUBRC <> 0.
      MESSAGE E000(ZY) WITH 'Unable to download file format 1'(003).
      V_REP1 = 1.
    ELSE.
*-- Success download
*-- Move date and time into variable
      MOVE : SY-DATUM TO V_DATE,
             SY-UZEIT TO V_TIME.

  - Update Batch number into ZOUTBOUND1 for Success download file
      LOOP AT I_TMPDATA1.

        UPDATE ZOUTBOUND1
           SET : BATCH_NO    = V_BATCH_NO ,
                 BATCH_DATE  = V_DATE
                 BATCH_TIME  = V_TIME
         WHERE ITEM NO       = I TMPDATA1-ITEM NO.

        IF SY-SUBRC <> 0.
          MESSAGE E000(ZY) WITH
          'Unable to update batch number into ZOUTBOUND1 table'(014).
        ENDIF.

      ENDLOOP.

    Update ZBATCH_CONTROL_O by update time and date
      UPDATE ZBATCH_CONTROL_O
         SET : BATCH_DATE = V_DATE ,
               BATCH_TIME = V_TIME
       WHERE BATCH NO    = V BATCH NO
      IF SY-SUBRC <> 0.
        MESSAGE E000(ZY) WITH
        'Unable to update ZBATCH_CONTROL_O table in format 1'(009).
      ENDIF.

    ENDIF.  " End check sy-subrc for WS_DOWNLOAD
  ENDIF.  " End if V LINES1 > 0

*-- Check for download file format 3

  IF V_LINES3 > 0.

*-- Flag check download success = 0. unsuccess = 1.
    V REP3 = 0.

    CALL FUNCTION 'WS_DOWNLOAD'
      EXPORTING
        FILENAME  =  V_DEFAULT_NAME
        FILETYPE  =  C_DAT
      TABLES
        DATA TAB  =  I TMPDATA3.

    IF SY-SUBRC <> 0.
      MESSAGE E000(ZY) WITH 'Unable to download file format 3'(004).
      V_REP3 = 1.
    ELSE.
*-- Success download
*-- Move date and time into variable
      MOVE : SY-DATUM TO V_DATE,
             SY-UZEIT TO V TIME.

*-- Update Batch number into ZOUTBOUND3 for Success download file
      LOOP AT I_TMPDATA3.

        UPDATE ZOUTBOUND3
           SET : BATCH_NO   = V_BATCH_NO ,
                 BATCH_DATE = SY-DATUM   ,
                 BATCH_TIME = SY-UZEIT
         WHERE ITEM NO    = I TMPDATA3-ITEM NO.

        IF SY-SUBRC <> 0.
          MESSAGE E000(ZY) WITH
          'Unable to update batch number into ZOUTBOUND3 table'(016).
        ENDIF.

      ENDLOOP.

    Update ZBATCH_CONTROL_O by update time and date
      UPDATE ZBATCH_CONTROL_O
         SET : BATCH_DATE = V_DATE ,
               BATCH_TIME = V_TIME
       WHERE BATCH_NO   = V BATCH NO.

      IF SY-SUBRC <> 0.
        MESSAGE E000(ZY) WITH
        'Unable to update ZBATCH_CONTROL_O table in format 3'(015).
      ENDIF.

    ENDIF.
  ENDIF.

ENDFORM.                              " 03000 DATA INTERFACE

**********************************************************************
*      FORM 04000_PRINT_REPORT                                       *
 -----                                        -------------------*
*      Description: This Form for print report                       *
**********************************************************************
FORM 04000_PRINT_REPORT.

  WRITE : '*** SUCCESS RECORD ***'(005).

  IF V REP1 = 0.

    WRITE : /1 '*** File format 1 ***'(006).

    LOOP AT I_TMPDATA1.
      WRITE : /1  I_TMPDATA1-ITEM_NO,
              15  I_TMPDATA1-INST_DATE,
              35  I_TMPDATA1-INST_TIME,
              55  I TMPDATA1-GOODS_CODE.
    ENDLOOP.
  ENDIF.

  IF V_REP3 = 0.

    WRITE : /1 '*** File format 3 ***'(007).

    LOOP AT I_TMPDATA3.
      WRITE : /1  I_TMPDATA3-ITEM_NO,
              15  I_TMPDATA3-INST_DATE,
              35  I_TMPDATA3-INST_TIME,
              55  I TMPDATA3-GOODS CODE.
    ENDLOOP.
  ENDIF.

  SKIP.
  WRITE : /1 '*** ERROR RECORD ***'(008).
  SKIP.

  IF V REP1 = 1.

    WRITE : /1 '*** File format 1 ***'(006).

    LOOP AT I_TMPDATA1.
      WRITE : /1  I_TMPDATA1-ITEM_NO,
              15  I_TMPDATA1-INST_DATE,
              35  I_TMPDATA1-INST_TIME,
              55 I TMPDATA1-GOODS CODE.
    ENDLOOP.
  ENDIF.

  IF V_REP3 = 1.

    WRITE : /1 '*** File format 3 *** (007).

    LOOP AT I_TMPDATA3.
      WRITE : /1  I_TMPDATA3-ITEM_NO,
              15  I TMPDATA3-INST_DATE,
                 35  I_TMPDATA3-INST_TIME,
                 55 I_TMPDATA3-GOODS_CODE.
     ENDLOOP.
   ENDIF.

ENDFORM.                                  "04000_PRINT_REPORT
```

APPENDIX C: Source Code of Outbound Customized Application Integration Program for UNIX.

```
REPORT ZOUTBINT LINE-SIZE 120
                LINE-COUNT 65
                NO STANDARD PAGE HEADING.

*********************************************************************
*  Program               ZOUTBINT1
*  Author                Supaporn Wongwithit
*  Created               03/04/2002
*  Transport No.
*
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   Description         : Interface program between SAP and another
                          system.
                          This program will transfer data from SAP to
                          another system. This program will retrieve
                          data from SAP database in last half and hour.
                          And populate/reformat data assign batch no,
                          store data into Backup database. After that
                          generate Outbound file send to another system.

   Input

   Output              : Text file and report for success and
                          unsuccessful records.

   Called from

   Called to

   Includes

   Function Modules

   Logical Database

   High Level Design :

 ------------------------------------------------------------------

*********************************************************************
       TABLES
*********************************************************************
TABLES: ZOUTBOUND1 ,           " Outbound format 1
        ZOUTBOUND2 ,           " Outbound format 2
        ZOUTBOUND3 ,           " Outbound format 3
        ZBATCH_CONTROL_O.      " Outbound Batch Control

*********************************************************************
•      DATA                                                         *
*********************************************************************
 -------------- Constants - - - - - - - - - - - - - - - - - - - - - - - -
CONSTANTS: C_DEFAULT_PATH(50) TYPE C VALUE  /usr/'      , " Path unix
           C_FILENAME(50)     TYPE C VALUE 'OUTBOUND'   , " File name
```

```
            C_DAT(3)           TYPE C VALUE 'DAT'          DAT file
            C_TXT(4)           TYPE C VALUE '.TXT'         Txt file
            C_102(3)           TYPE C VALUE '102'        " Trans type
            C_107(3)           TYPE C VALUE '107'          Trans type
            C_108(3)           TYPE C VALUE '108'          Trans type
            C_109(3)           TYPE C VALUE '109'          Trans type
            C_111(3)           TYPE C VALUE '111'          Trans type
            C_112(3)           TYPE C VALUE '112'          Trans type
            C_113(3)           TYPE C VALUE '113'          Trans type
            C_114(3)           TYPE C VALUE '114'          Trans type
            C_116(3)           TYPE C VALUE '116'          Trans type
            C_COLON(1)         TYPE C VALUE ':'
            C 01               LIKE INRI-NRRANGENR
                                      VALUE '01'        , " Interval#
            C_ZOUTBATCHC       LIKE INRI-OBJECT
                                      VALUE 'ZOUTBATCHC'.   Obj#

*------- Working areas ---------------------------

*------- Internal table --------------------------
TYPES: BEGIN OF TYP_HEADER,
         HEADER(100)    TYPE C ,           " Header
       END OF TYP_HEADER.

TYPES: BEGIN OF TYP_I_MESSAGE,
         MSG_TEXT(250) TYPE C ,            " Message Text
       END OF TYP I MESSAGE.

DATA : I_TMPDATA1     LIKE ZOUTBOUND1    OCCURS  0  WITH HEADER LINE.
DATA : I_TMPDATA2     LIKE ZOUTBOUND2    OCCURS  0  WITH HEADER LINE.
DATA : I_TMPDATA3     LIKE ZOUTBOUND3    OCCURS  0  WITH HEADER LINE.
DATA : I_HEADER       TYPE TYP_HEADER    OCCURS  0  WITH HEADER LINE.
DATA : I MESSAGE      TYPE TYP I MESSAGE OCCURS  0  WITH HEADER LINE.

*  ----- Variable ----------------------------------

DATA : V_REPID          LIKE SY-REPID,    Report-id
       V_LINES1         TYPE N ,          Lines from format file 1
       V_LINES2         TYPE N ,          Lines from format file 2
       V LINES3         TYPE N ,          Lines from format file 3
       V_SUMLINES       TYPE N ,          Sum Lines from format file
       V_CSUMLINES(6)   TYPE C ,          Char for Sum lines of file
       V_REP1           TYPE N ,          Flag chk for report type 1
       V_REP3           TYPE N ,          Flag chk for report type 3
       V_DATE           LIKE SY-DATUM ,   System Date
       V_TIME           LIKE SY-UZEIT     System Time
       V_BATCH_NO       LIKE ZBATCH_CONTROL 0-BATCHNO , " Batch#
       V_DEFAULT_NAME   LIKE RLGRAP-FILENAME , " File Name
       V MSG TEXT(250)  TYPE C.                 " Message text

*********************************************************************
       SELECTION   SCREEN                                           *
*********************************************************************
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.

PARAMETERS: P PATH(50) TYPE C DEFAULT '/usr/'. " Path unix

PARAMETERS: P_TEST AS CHECKBOX default 'X'.     " Test mode chk box

SELECTION-SCREEN END OF BLOCK B1.
```

```
*********************************************************************
       I NITIALIZATION
*********************************************************************
INITIALIZATION.

MOVE SY-REPID TO V REPID.

*********************************************************************
       T O P   O F  PAGE
*********************************************************************
TOP-OF-PAGE.

* -- Call standard report heading
  SUMMARY.
  FORMAT COLOR COL HEADING.

* -- Write header report
  PERFORM   HEADER(ZREPHEAD) USING
                              V_REPID                  " report-id
                              SY-LINSZ.                " line-size

  WRITE : /1   'ITEM NO'(010),
           15  'INSTUCTION DATE'(011),
           35  'INSTUCTION TIME'(012),
           55  'GOODS CODE'(013).

*********************************************************************
•      BEGIN    SELECTION                                          *
*********************************************************************
START-OF-SELECTION.

*-- For test only Prototype Phase
  IF P_TEST = 'X'.

*-- Get data from SAP table
    PERFORM 01000_GET_DATA.

  - Check internal table before process
    IF V SUMLINES > 0.

*-- Path file and file name.
      CONCATENATE P_PATH C_FILENAME SY-DATUM SY-UZEIT C TXT
             INTO V_DEFAULT_NAME.

*-- Generate Batch number
      PERFORM 02000_GEN_BATCH NUMBER.

*-- Interface data : Send data from SAP to another system
      PERFORM 03000_DATA_INTERFACE.

*-- Print Report
      PERFORM 04000_PRINT_REPORT.

    ELSE.
      MESSAGE S000(ZY) WITH
      'No data Selected!!!'(002).
    ENDIF.

  ENDIF.
```

```
END-OF-SELECTION.

**********************************************************************
*      END   OF   SELECTION                                          *
**********************************************************************

**********************************************************************
       BEGIN     FORMS
**********************************************************************

**********************************************************************
       FORM 01000 GET DATA

       Description: Select data from customized table that have data*
                    from user exit and this data will be transfer to*
                    text file and send to another system by using   *
                    internal table I_DATA.
**********************************************************************
FORM 01000_GET_DATA.

*-- Clear internal table
  REFRESH : I_TMPDATA1,
            I_TMPDATA3.

  CLEAR   : I_TMPDATA1,
            I_TMPDATA3,
            V LINES1  ,
            V_LINES2  ,
            V_LINES3
            V_SUMLINES,
            V CSUMLINES.

*-- Select data from SAP table and insert into internal table before
*-- send it out to another system.

  SELECT *
    INTO TABLE I_TMPDATA1
    FROM ZOUTBOUND1
   WHERE BATCH NO = ''

  DESCRIBE TABLE I TMPDATA1 LINES V LINES1.

  SELECT *
    INTO TABLE I_TMPDATA3
    FROM ZOUTBOUND3
   WHERE BATCH NO = ''

  DESCRIBE TABLE I TMPDATA3 LINES V LINES3.

  V SUMLINES = V LINES1 + V LINES2 + V LINES3.

*-- should add some where clause for get only last 1/2 hr. for send
 t
*-- out.

ENDFORM.                                " 01000_GET_DATA

**********************************************************************
       FORM 02000 GEN BATCH NUMBER
```

```
*-------------------------------------------------------------------*
       Description: This Form will generate batch number and update *
                    batch number into ZOUTBOUNDX internal table for *
                    each file format
*********************************************************************
FORM 02000_GEN_BATCH_NUMBER.

  REFRESH : I HEADER  .

  CLEAR   : I_HEADER
            V_BATCH_NO,
            V_DATE
            V_TIME

  CALL FUNCTION 'NUMBER_GET_NEXT'
    EXPORTING
      NR_RANGE_NR                    = C_01
      OBJECT                         = C_ZOUTBATCHC
    IMPORTING
      NUMBER                         = V_BATCH_NO
    EXCEPTIONS
      INTERVAL NOT_ FOUND            = 1
      NUMBER_RANGE_NOT_INTERN        = 2
      OTHERS                         = 8.

    IF SY-SUBRC <> 0.
      MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
      WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
    ELSE.
      MOVE : SY-DATUM TO V_DATE,
             SY-UZEIT TO V_TIME.

    Update batch number into ZBATCH CONTROL_O.
      ZBATCH CONTROL 0-BATCH NO = V BATCH_NO.

      INSERT ZBATCH_CONTROL_O.

      WRITE V_SUMLINES TO V_CSUMLINES.
*-- File header example : 2002021811:20:0100005201500016962
      CONCATENATE V_DATE V_TIME+0(2) C_COLON
                  V TIME+2(2) C_COLON V TIME+4(2) V CSUMLINES
V_BATCH_NO
                  INTO I_HEADER-HEADER.

      APPEND I_HEADER.
    ENDIF.

ENDFORM.                          " 02000 GEN BATCH NUMBER

*********************************************************************
       FORM 03000_DATA_INTERFACE
*-------------------------------------------------------------------
       Description: This Form will send data out from SAP to another*
                    System
*********************************************************************
FORM 03000_DATA_INTERFACE.

  OPEN DATASET V_DEFAULT_NAME FOR APPENDING IN TEXT MODE
                                       MESSAGE V_MSG_TEXT.
  IF SY-SUBRC <> 0.
    CONCATENATE 'File cannot be opened : ' V_MSG_TEXT INTO
                                    I MESSAGE-MSG TEXT.
    APPEND I_MESSAGE.
    CLOSE DATASET V_DEFAULT_NAME.
    EXIT.
  ENDIF.

*-- For UNIX
  OPEN DATASET V_DEFAULT_NAME FOR APPENDING IN TEXT MODE
                                        MESSAGE V MSG TEXT.


  IF SY-SUBRC <> 0.
    MESSAGE E000(ZY) WITH Unable to download header'(002).
  ENDIF.

  LOOP AT I_HEADER.
    TRANSFER I_HEADER TO V_DEFAULT_NAME.
  ENDLOOP.

  CLOSE DATASET V_DEFAULT_NAME.

*-- Check for download file format 1
  IF V_LINES1 > 0.

*-- For UNIX
    OPEN DATASET V DEFAULT NAME FOR APPENDING IN TEXT MODE
                                        MESSAGE V_MSG_TEXT.
    LOOP AT I_TMPDATA1.
      TRANSFER I_TMPDATA1 TO V_DEFAULT_NAME.
    ENDLOOP.

    CLOSE DATASET V_DEFAULT_NAME.

*-- Flag check download success = 0. unsuccess = 1.
    V REP1 = 0.

    IF SY-SUBRC <> 0.
      MESSAGE E000(ZY) WITH 'Unable to download file format 1'(003).
      V_REP1 = 1.
    ELSE.
    Success download
    Move date and time into variable
      MOVE : SY-DATUM TO V_DATE,
             SY-UZEIT TO V_TIME.

*-- Update Batch number into ZOUTBOUND1 for Success download file
      LOOP AT I TMPDATA1.

        UPDATE ZOUTBOUND1
           SET : BATCH_NO   = V BATCH_NO ,
                 BATCH_DATE = V_DATE
                 BATCH_TIME = V_TIME
         WHERE ITEM NO      = I TMPDATA1-ITEM NO.

        IF SY-SUBRC <> 0.
          MESSAGE E000(ZY) WITH
          'Unable to update batch number into ZOUTBOUND1 table'(014).
        ENDIF.
      ENDLOOP.

*-- Update ZBATCH_CONTROL_O by update time and date
      UPDATE ZBATCH_CONTROL_O
         SET : BATCH_DATE = V_DATE ,
               BATCH_TIME = V_TIME
       WHERE BATCH NO    = V BATCH NO.

      IF SY-SUBRC <> 0.
        MESSAGE E000(ZY) WITH
        'Unable to update ZBATCH_CONTROL_O table in format 1'(009).
      ENDIF.

    ENDIF.  " End check sy-subrc for WS_DOWNLOAD
  ENDIF.  " End if V LINES1 > 0

  - Check for download file format 3

  IF V_LINES3 > 0.
    OPEN DATASET V_DEFAULT_NAME FOR APPENDING IN TEXT MODE
                                               MESSAGE V MSG TEXT.

*-- Flag check download success = 0. unsuccess = 1.
    V_REP3 = 0.

    IF SY-SUBRC <> 0.
      MESSAGE E000(ZY) WITH 'Unable to download file format 3'(004).
      V_REP3 = 1.
    ELSE.
*-- Success download
*-- Move date and time into variable
      MOVE : SY-DATUM TO V_DATE,
             SY-UZEIT TO V_TIME.

    Update Batch number into ZOUTBOUND3 for Success download file
      LOOP AT I_TMPDATA3.

        UPDATE ZOUTBOUND3
           SET : BATCH_NO   = V_BATCH_NO ,
                 BATCH_DATE = SY-DATUM   ,
                 BATCH_TIME = SY-UZEIT
         WHERE ITEM NO     = I TMPDATA3-ITEM NO.

        IF SY-SUBRC <> 0.
          MESSAGE E000(ZY) WITH
          'Unable to update batch number into ZOUTBOUND3 table'(016).
        ENDIF.

      ENDLOOP.

*-- Update ZBATCH_CONTROL_O by update time and date
      UPDATE ZBATCH_CONTROL_O
         SET : BATCH_DATE = V_DATE ,
               BATCH_TIME = V_TIME
       WHERE BATCH_NO    = V_BATCH_NO.

      IF SY-SUBRC <> 0.
        MESSAGE E000(ZY) WITH
        'Unable to update ZBATCH_CONTROL_O table in format 3'(015).
      ENDIF.

    ENDIF.
    CLOSE DATASET V DEFAULT NAME.

  ENDIF.

ENDFORM.                               " 03000 DATA INTERFACE

**********************************************************************
       FORM 04000 PRINT REPORT

       Description: This Form for print report
**********************************************************************
FORM 04000_PRINT_REPORT.

  WRITE : '*** SUCCESS RECORD ***'(005).

  IF V REP1 = 0.

    WRITE : /1 '*** File format 1 ***'(006).

    LOOP AT I_TMPDATA1.
      WRITE : /1  I_TMPDATA1-ITEM_NO,
              15  I_TMPDATA1-INST_DATE,
              35  I_TMPDATA1-INST_TIME,
              55  I TMPDATA1-GOODS CODE.
    ENDLOOP.
  ENDIF.

  IF V REP3 = 0.

    WRITE : /1 '*** File format 3 ***'(007).

    LOOP AT I_TMPDATA3.
      WRITE : /1  I_TMPDATA3-ITEM_NO,
              15  I_TMPDATA3-INST_DATE,
              35  I_TMPDATA3-INST_TIME,
              55  I TMPDATA3-GOODS CODE.
    ENDLOOP.
  ENDIF.

  SKIP.
  WRITE : /1 '*** ERROR RECORD ***'(008).
  SKIP.

  IF V REP1 = 1.

    WRITE : /1 '*** File format 1 ***'(006).

    LOOP AT I TMPDATA1.
      WRITE : /1  I_TMPDATA1-ITEM_NO,
              15  I_TMPDATA1-INST_DATE,
              35  I_TMPDATA1-INST_TIME,
              55  I TMPDATA1-GOODS CODE.
    ENDLOOP.
  ENDIF.

  IF V REP3 = 1.
```

```
    WRITE : /1 '*** File format 3 ***'(007).

    LOOP AT I_TMPDATA3.
      WRITE : /1  I_TMPDATA3-ITEM_NO,
              15  I_TMPDATA3-INST_DATE,
              35  I_TMPDATA3-INST_TIME,
              55 I TMPDATA3-GOODS CODE.
    ENDLOOP.
  ENDIF.

  WRITE : /1 '*** MESSAGE ***'(017).

  LOOP AT I_MESSAGE.
    WRITE : /1 I_MESSAGE-MSG_TEXT.
  ENDLOOP.
ENDFORM.                          " 04000_PRINT_REPORT
```