# SAP CHANGE MANAGEMENT

---

A Project

Presented

to the Faculty of

California State University, Chico

---

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Arvind Sureshchandran Nair

Spring 2009

SAP CHANGE MANAGEMENT

A Project

by

Arvind Sureshchandran Nair

Spring 2009

APPROVED BY THE DEAN OF THE SCHOOL OF
GRADUATE, INTERNATIONAL, AND INTERDISCIPLINARY STUDIES:

Susan E. Place, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

Abdel-Moaty M. Fayek
Graduate Coordinator

Seung-Bae Im, Ph.D., Chair

James R. Connolly, Ph.D.

# TABLE OF CONTENTS

PAGE

# LIST OF FIGURES

FIGURE PAGE

# ABSTRACT

## SAP CHANGE MANAGEMENT

by

Arvind Sureshchandran Nair

Master of Science in Computer Science

California State University, Chico

Spring 2009

SAP systems are used in organization to facilitate their business. It provides seamless integration of business functions. SAP is configurable to suit the business needs of the company. SAP customer can also develop programs, reports to mold SAP to function per the business requirements. All the changes in terms of developments and customizations create change requests, also known as transport requests. The transports are essentially made in a development system and moved to the production system. Since business requirements changes on a continual basis, there would be many transport requests that go from development system to production system. The regulation of the transport requests is known as change management.

Organization using SAP systems develop many processes and procedures to manage changes. These often result in manual documentation of the changes, approvals tracking and manual transport management. The auditors and organization's

management also need reports showing the amount of changes in the system. A huge effort is spend every day on change management to document every step of the request. In addition, all the approvals have to be maintained for audit reasons and proper review and testing of the changes in the system should be performed.

The purpose of this project is to develop a system that can automate many tasks in change management. It aims to simplify the process of tracking business requirements, developments, configurations, testing and validation. The system ensures that it has proper approval before a certain change is moved to production. The system has the capability to make sure there is security at each stage of the change request. It also provides reporting to the management as well as auditors.

The system incorporates various building blocks of SAP's proprietary language, ABAP, SAP tools like SAPconnect, Transport Management System and Change and Transport Organizer. SAP standard workflow is triggered at various stages for notification. SAP authorizations are used to secure the system so that changes do not get into production without appropriate approval. This is key to change management since auditors monitor proper segregation of duties. The system also gathers all the documentation that should be associated with a change request. The system uses standard function modules provided by SAP to ensure smooth transport management.

# CHAPTER I

# INTRODUCTION

Systems, Applications and Products in Data Processing (SAP) is the world’s leading vendor of business application software. SAP provides the basic business functionalities at the same time leaving room for adding new functionalities and enhancements. Any new input that is made in the system updates logically linked modules which help business react to the immediate information and changes. SAP has two sides to its software, the application or the functional side of the software and basis or the technical side. The functional side deals with the applications such as human resources, personnel planning, materials management, investment management and so on. The technical side is mainly concerned with operating system, database, SAP kernel and middleware tools. The group of people that works on the functional side is known as functional analysts and those on the technical side are known as ‘basis’ or technical analysts [5].

SAP systems have a concept of a “System Landscape” [5]. System landscape typically comprises of a development, quality assurance (QA) and production environments. All SAP customizations and developments are done in the development system. The developments and customizations are then moved into quality assurance environment for testing. After successful testing, it is then moved into production environment. This movement of customization, developments and modifications are made possible by SAP's Transport Management System (TMS) [4], [5], [7].

In the Figure 1, transport routes are defined in the TMS which ensures that changes that are released from development system (R3D) will go to the quality assurance system (R3Q) and then to the production system (R3P). The changes released from development system follow the transport routes (ZR3D and SAP) to QA system. These transport routes ensure that any change that is released from the development system will always go to QA system. The transport route that connects a development system and QA system is called consolidation route. The transport route that connects a QA system and a production system is called delivery route. The changes reside in the respective system's buffer. Then, the changes are imported into the systems. The import can be done manually or automatically through a batch job. The change is stored in the form of a transport. A transport is defined as a package of SAP objects that is transferred from one system to another. A transport is identified by ten alphanumeric letters that

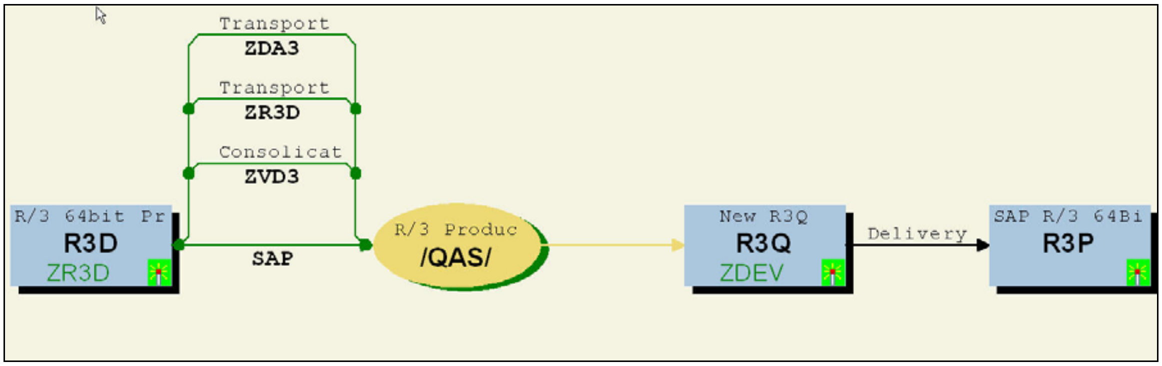
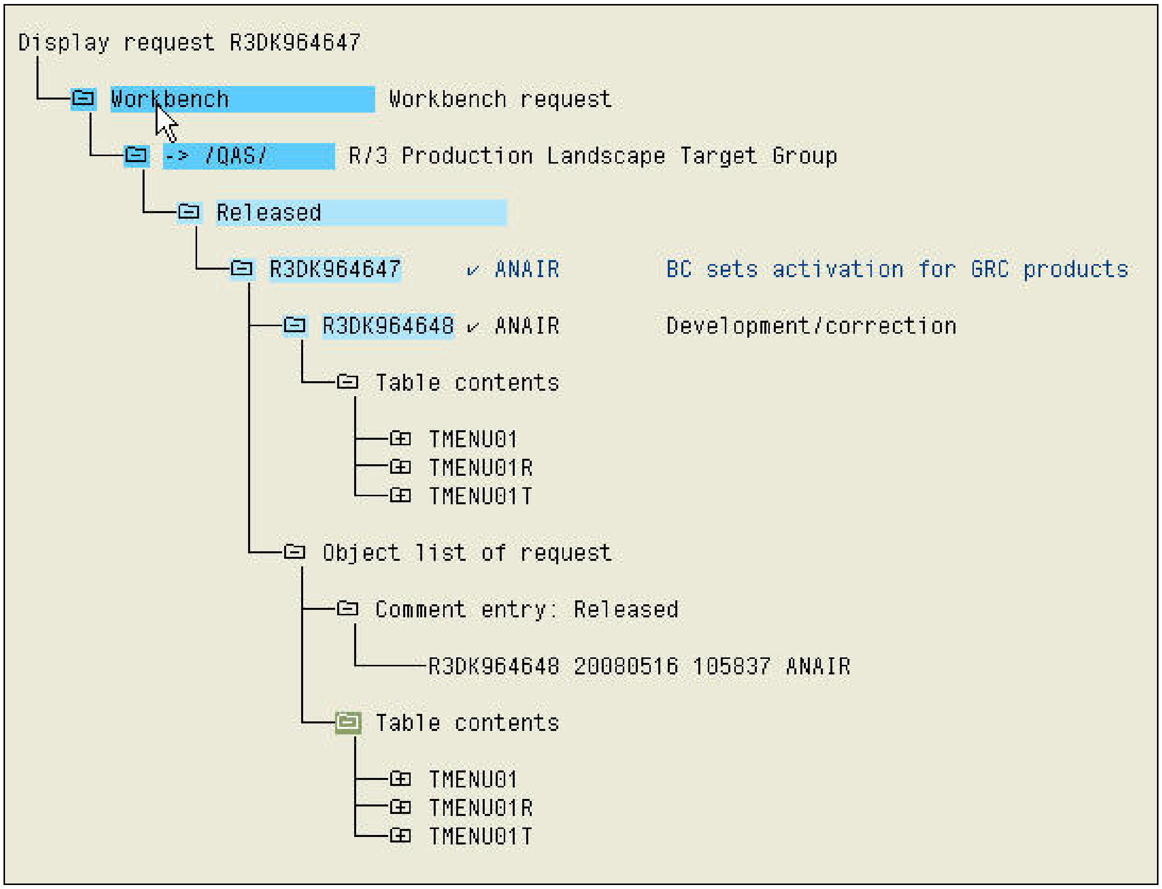


Fig. 1. SAP landscape (systems and transport routes).

starts with the development system identifier (R3D, in this case). The system identifier is then followed by letter ‘K’ and then six digits, for example, R3DK964647.

This transport number locks a complete of changes to SAP system objects. The locks on those objects are released only when the transport is released to the QA system. In Figure 2, the transport number R3DK964647 is called the header of the transport and the number R3DK964648 is called task of the transport. A transport header can contain a number of tasks.



```
Display request R3DK964647
 └─ Workbench              Workbench request
     └─ -> /QAS/           R/3 Production Landscape Target Group
         └─ Released
             └─ R3DK964647      ✓ ANAIR        BC sets activation for GRC products
                 ├─ R3DK964648 ✓ ANAIR        Development/correction
                 │   └─ Table contents
                 │       ├─ TMENU01
                 │       ├─ TMENU01R
                 │       └─ TMENU01T
                 └─ Object list of request
                     ├─ Comment entry: Released
                     │   └─────R3DK964648 20080516 105837 ANAIR
                     └─ Table contents
                         ├─ TMENU01
                         ├─ TMENU01R
                         └─ TMENU01T
```

Fig. 2. Objects contained in a transport.

A change in SAP is business critical; hence, there should be proper management of transports. The lifecycle of a transport from development system till production system should be done through adequate testing and relevant approval. A business user identifies a defect or modification to current transaction or a report. A developer makes appropriate modification and releases a change from a development system. This change has to be approved by one of the analysts. The change is then moved to QA. The user tests the changes and approves it. The change should then be approved by the process owner. All such changes are then reviewed by a group of functional analysts to check for any dependencies in other business processes and then approve the list for production import. The basis personnel will then move the changes to production.

## Problem

The process from the point where user requests for a modification or reports a bug till transport to production is done through email approvals. Over a period of time the changes that are made are not traceable. The management does not have visibility to the changes. Also it becomes difficult to revisit the changes that have gone to production. In case of huge projects and implementations, the number of changes is very high in number. The project involves a group of developers and analysts. Each developer or analyst will have a number of transports. In such case, there is a need for change control such that all bugs can be traced along with projects. Also these can be reviewed on request.

Companies in the United States that are under the jurisdiction of US Securities and Exchange Commission have to comply with Sarbanes-Oxley (SOX) [1], [2], [3].

SOX policies strive to prevent wrongdoing and improve corporate administration. The companies publish their financial information on the stock markets along with their profits and trades. SOX ensure that all the information is correct, all procedures, administrative activities are followed as per documented procedures. External auditors review all the activities. These activities not only include all the administrative procedures but also all IT processes that affect business. Hence, all changes that are made by analysts and developers are checked by auditors. These changes are reviewed by auditors annually in order for companies to comply with SOX.

## Purpose

The changes that are made in an SAP system in a year become totally untraceable since every request, approval emails have to be provided to auditors. The management therefore wants a system to track all changes that start from development system and end in Production. The following are issues that the management wants to address:

- Development done by developers should not be released by them. The changes should be only released from the development system by the manager.
- The system should be able to account for all daily changes as well as big projects.
- It should have SOX approval for all SOX relevant changes.
- Management should have the ability to approve changes for production.
- All data relevant to the changes should be captured in the same change management which includes.

- All testing data (unit, integration and user acceptance).
- Impact Analysis.
- SOX relevance.
- Ability to move the transport without manual intervention except production.

SAP has a Product Lifecycle Management [7] product which gives basic document management and audit management. But in order to have complete change management system that integrates well with TMS, a change management tool will be developed that will address the issues listed above. It will give the auditors to find all the information that they need about Production changes. Also, it will give management a clear view of the set of changes that has happened during any period of time.

# CHAPTER II

## LITERATURE REVIEW

The change management system is built on top of the transport management system. The idea of change management became more important with the advent of Sarbanes-Oxley (SOX) compliance [1], [2], [3]. Hence to first address change management, it is important to understand what Sarbanes-Oxley meant. Since it is a control that applies to all departments in a company that directly or indirectly affects finance, an understanding of what is expected is important. This will give requirements gathering a head start. “Using a single software platform/electronic repository to coordinate and document all controls, documentation, test results and other relevant support material is useful for several reasons.” [1]

The quote from *A Complete Guide to Sarbanes-Oxley* by Stephen M. Bainbridge gives some insight into what audit requirements would be. It emphasizes the need for software to aggregate all the relevant documents that would be required. It is very important to set controls that can be used to validate the system. SOX controls act as guidelines for auditors to validate systems. A single repository with all change documents present a few advantages such as:

- Reduces external audit consulting hours and hence saves money since every change is documented per agreed controls.

- Regulates change control.
- Ease of reporting.

The impact of SOX on IT processes is as high as it is on financials. Hence, it is very important for management to consider SOX impact on all these processes, hence have proper SOX controls.

- Change requests are submitted on an appropriate form and are reviewed and approved by Management.
- Change are prioritized and tracked by management through systematic or manual procedures.
- All changes will be tested in a controlled test environment prior to implementation
- Changes to production environments are approved prior to implementation. This includes developer and user testing. [2, p. 50]

These are among the few controls that are laid by SOX which calls for change management. It advises management to review all the changes that are submitted to ensure that there is proper review by a group who is responsible for the company's financial standing. Testing changes before going to production is one of most important factor since it catches any issues which might potentially impact the system which may cause financial loss for example, a downtime of the system during financial year-end close. Hence, change management should ensure that the following are taken into account when moving changes to production:

- Has the management approved the changes to be implemented?
- Has the changed been tested by multiple users who are involved in the change that will go to production?
- Has the final review of the change been done before they make it to production

It is also important to understand what types of changes need to be considered for change management.

The following types of changes are often considered during daily change review meetings:

- Emergencies, releases, fixes etc. These are normally related to circumstances in which production is down.
- Database clones, restores, links, or new instances. [3]

All the changes are not directly provided by SAP hence there is a need for customization. It not only explains that changes in take place SAP landscape are reviewed by the review board but any system change that is likely to affect the production system is also reviewed. Hence, this helps the overall design to consider not just the customizations and developments but also relevant system changes as well. This presented a deviation from the norm. The changes in an SAP system always associates to a transport but in case of system change, for example, an SAP parameter change, will not create a transport hence such changes need to be accounted for which presents a small challenge.

Another important point to consider during design is the amount of documentation that should be attached to each change request. The internal auditors in the company need to validate the system before external auditors come for audit. They will need to verify all the processes in the system and ensure that it complies with existing regulations. Since SAP is an integrated environment, auditors typically ensure that all changes that go to production are properly documented. Hence, it is imperative to consider how much relevant documentation needs to be attached to support a certain change request. The following explains transport movement in a landscape:

> The implementation of R/3 requires, at the bare minimum, that you customize the R/3 system using the IMG. Customizing is performed in a client in the development system; transferred to a client in the quality assurance system; and, finally made available to a client in the production system. Similarly, changes due to development in the ABAP Workbench require organized techniques of distribution from the development system to the quality assurance system and the production system. [4]

The customizing and development changes are made in a development system and then moved to the quality assurance system and finally after testing they are moved in to the production system. This is the main idea of change management in SAP. Although, these changes can be done in each system manually but in order to maintain control of all the changes that go to production, changes are moved from one system to another by means of Transport Management System. The main issue with the transport management system is that it is a GUI transaction in which case, the admin has to manually import approved transports. In order to have seamless movement of transport through the application, this process should be automated.

> The ‘tp’, transport control program allows the system or transport administrator to perform all the management functions for the transport system. The program “tp” includes functions for exporting, importing, buffer actions, managing disk space of the transport system. [5, pp. 273-286]

The above explains how transports can be managed at the SAP level as well as the operating system level. Hence, this helps to automate transport movement from one system to another in the landscape. This is particularly important because the SAP transaction STMS that is used to manage the transport management system is a dialog-based transaction and hence can be used manually to move transports. Since we need to automate change management process, this step becomes critical enough that it cannot be used to automate the process. The use of ‘tp’ helps the design in a very significant manner. It can be run to import changes into the system automatically when released from the development system. “tp” can be called directly with appropriate options at the operating system level. This allows transports to be automated. The program can be used for all transport administrative tasks.

After the design, it is important to consider the building blocks of the application. ABAP provides a wide variety of elements, data types, structures and control elements to build an application. ABAP provides all the fundamental data types and controls flow like any traditional language.

> Development of Dialog Transactions. The dialog program initially consists of components that describe the program user interface. These programs are called dynpros in ABAP/4 (Advanced Business Application Programming Language / 4th Generation). A dynpro (= dynamic program) is a screen (= dialog mask = dialog) with an associated flow control. The dialog masks and flow control are processed in the Screen Painter (development tool). With regard to the screen layout, this procedure can be compared with the dialog editors of other development environments, such as Visual Basic and Visual C++. [6, pp. 183-187]

During design of the elements, it is important to consider how interactive the software should be made and what language elements should be used to build it. Should the complete application be developed on the SAP GUI environment or should it be given a web-look? Who are the users of the system? Hence, answers to such questions will help decide the way in the user interface should be built. ABAP also provides screen programming for providing good user interfaces. The Screen painter tool can be used to build the screens as well the flow control. SAP applications are highly integrated and run on a common platform.

> Building on a Framework. The most advanced step is to use the composite application framework and services from SAP R/3 to create entire new applications that build on everything that’s gone before.

> Extending SAP R/3 creates services that allow you to reuse data and functionality between applications. [7, pp. 323-332]

This framework allows development of applications and tools that aid automation of functions. SAP provides a lot of function modules and application program interfaces that can be reused to build new applications. The project too reuses existing functionalities and lays its new functionality around them. The change management is based on transport management system of SAP hence will reuse many of its functions to achieve the desired results.

To associate the processing block to an event:

> By default, the event start-of-selection is attached to all events in ABAP/4. In your programs you can define a processing block and attach this block to an event keyword.
>
> For generally easy-to-read code, it is good practice to define sequential processing blocks in the order by which they will most likely be triggered during selection screen execution. It is also good practice to utilize the most important events for selection screen programming. These events are as follows:
>
> - The initialization event.
> - The at selection-screen event.
> - The at user-command event. [8, pp. 632-700]

Dialog programs have selection screens and they associated to events that are triggered when invoked. Certain practices should be followed when writing blocks of code that will be called when and event is called. When the selection screen is executed, there will be a series of events triggered which in turn will call its corresponding execution blocks. These are the building blocks of the user interface in the project.

> Dialog Applications. A report does not have enough functionality to create complete, interactive applications. The R/3 System has provided dialog-oriented applications for this purpose. Like reports, the application's basic structure is predefined, and the ABAP statements used here are at a lower level than this basic structure. Dialog-oriented applications are primarily based on the use of screen templates called screen programs in the R/3 system. [9, pp. 278-301]

The extract above from talks about the lack of functionality of ABAP reports to create interactive applications. It talks about the screen programs that provide this functionality. All dialog-based application call screens are event triggered. The screens again are programmed in ABAP and have distinct blocks of execution unit for each event.

In addition to all the configurations and developments for the tool, there had to appropriate security. It is important that following pieces of security are implemented:

- Developers cannot release the transport to QA. Only managers and designated analysts can.
- Requestor and Approver approvals can be modified by non- other than those mentioned on the change request.
- Only Change Review Board can approve the transport to Production.

Hence appropriate authorization objects should be created and checked appropriately in the code. This will also comply with SOX security requirements for the software.

# CHAPTER III

## REVIEW OF TOOLS AND TECHNOLOGIES

The SAP change management tool is written in Advanced Business Application Programming (ABAP) language. ABAP is SAP’s proprietary language. ABAP has all the constructs of a programming language. ABAP can be developed on SAP’s ABAP Workbench. ABAP Workbench has all the tools required for development of programs, function modules, screen elements, web services, database tables, structures etc. It has tight integration with Transport Organizer. ‘tp’, the transport control program is written in C. It is used to move transports from one system to another at the operating system level. The Transport Management System (TMS) calls this program when TMS functions are called from SAP GUI. The following tools and technologies used in the project can be explained in detail.

- Advanced Business Application Programming (ABAP) language [6], [8], [9]
- ABAP Workbench and Customizing [5], [6], [7], [8], [9]
- Change and transport System (CTS) [4], [5]
    - Change and Transport Organizer (CTO)
    - Transport Management System (TMS)
- SAPConnect [5]

## Advanced Business Programming Language

Many of the applications in SAP are written in ABAP. It is a 4GL (fourth generation) programming language that is proprietary to SAP. ABAP programs are compiled and executed within the SAP Basis. SAP Basis is a set of programs that interact with SAP's kernel modules [5] and provide all the functions for the SAP application. SAP kernel is a set of modules written in C++ that provides hardware and database abstraction for the SAP Basis. ABAP programs interact with the operating system, the database and also with SAP presentation (SAP GUI). Users logon to the SAP by means of SAP GUI or web browser.

ABAP programs are the base of the entire SAP Basis. The applications as well as services interacting with SAP kernel are written in ABAP. ABAP has a structure that is modular and each such structure is called a processing block. It has the following blocks as shown in Figure 3.

- Global data declaration: The variables and constants that are used in the current program, function module (FM) [5], [6], or report forms the starting block.
- Selection screen for input [6]: In case, the program has fields for user input, it is declared in this section. This is default event block.
- Dialog processing modules [5], [6], [9]: These modules are processing units before and after selection screen.
- Function Modules: These are reusable program units.
- Subroutines: These are used internal to a program or report.
- Reports/Programs: Programs or reports are main unit of executable code.
- Package [6]: Object that encapsulates ABAP objects.

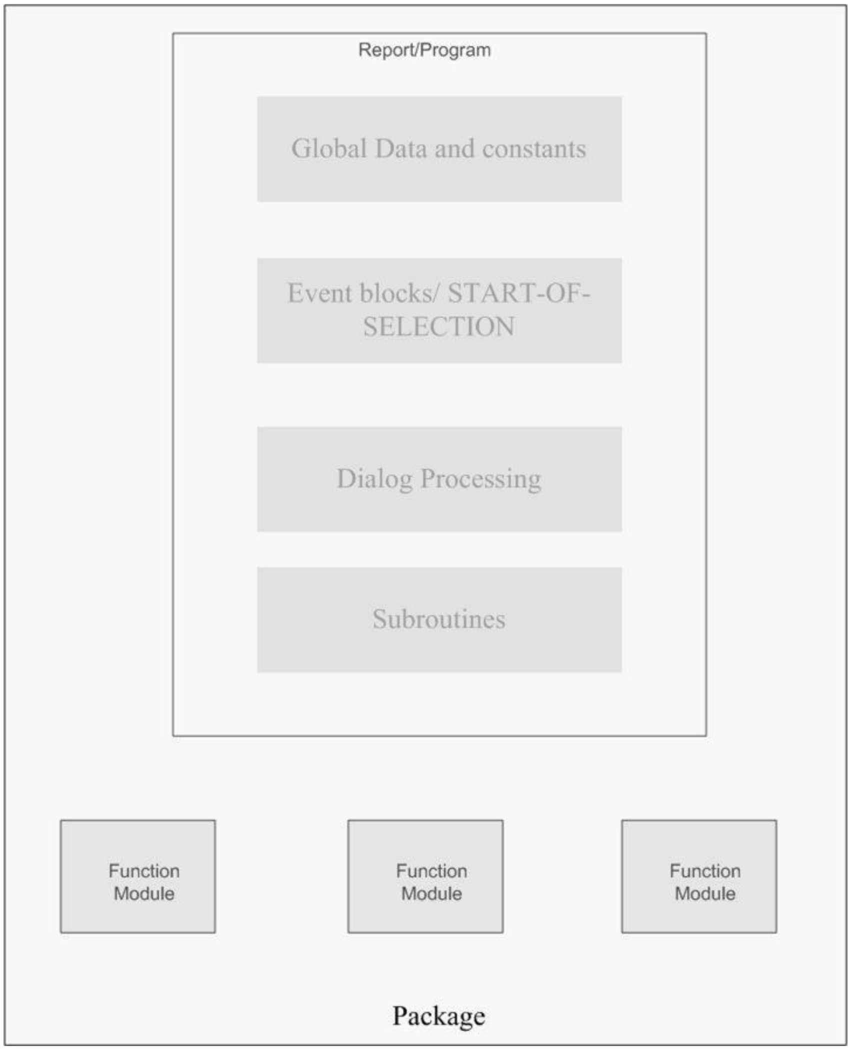


Fig. 3. ABAP language elements.

## ABAP Workbench and Customizing

ABAP has been used to build many portions of an SAP application. The modules are developed such that SAP customers can build on top of the applications to suit their company requirements. ABAP Workbench consists of all the tools to develop programs in ABAP. The ABAP Workbench is a set of tools, functions, programming methodologies, transport system and data dictionary that are fully integrated. It tracks changes through the CTS system which can be transferred to any other SAP system. It's tightly coupled with the data dictionary, transport system, class modeler, package builder, function/program builder, ABAP debugger, web publisher etc.

ABAP workbench can be entered through a central point, known as the Object Navigator. It is used to organize all the objects in the environment. It has the following sections as shown in Figure 4.

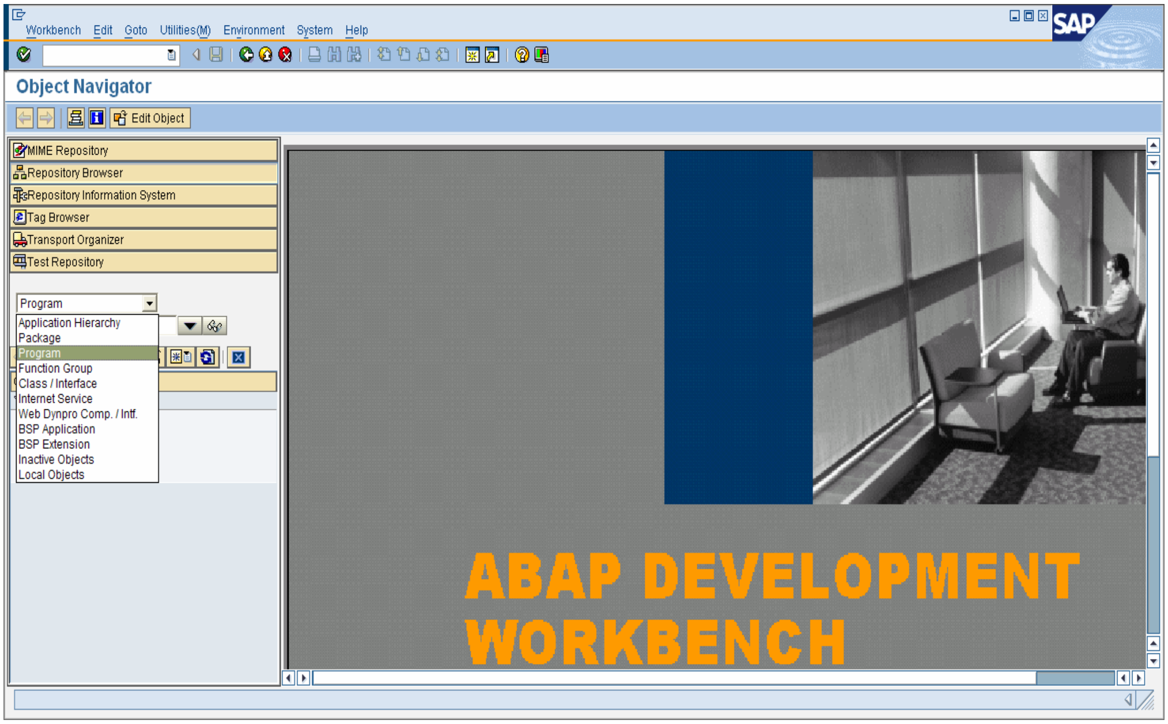


Fig. 4. ABAP workbench.

- ABAP Workbench tools [5]
- Object list
- Navigational Area

## Change and transport System

Every change that goes from one SAP system to another is governed by the Change and Transport System (CTS). All the development objects developed by means of the ABAP workbench, all the changes made to the repository objects such as tables, view etc. and the customization changes are managed by CTS. It comprises of mainly two sub-systems:

- Change and Transport Organizer (CTO): It mainly registers all developments and customizations in the system. Whenever a user makes any changes to its development objects i.e. programs, reports, table structures etc the system will maintain the version of the change through a transport request. Figure 5 shows the transport organizer where change request can be created, changed and displayed.

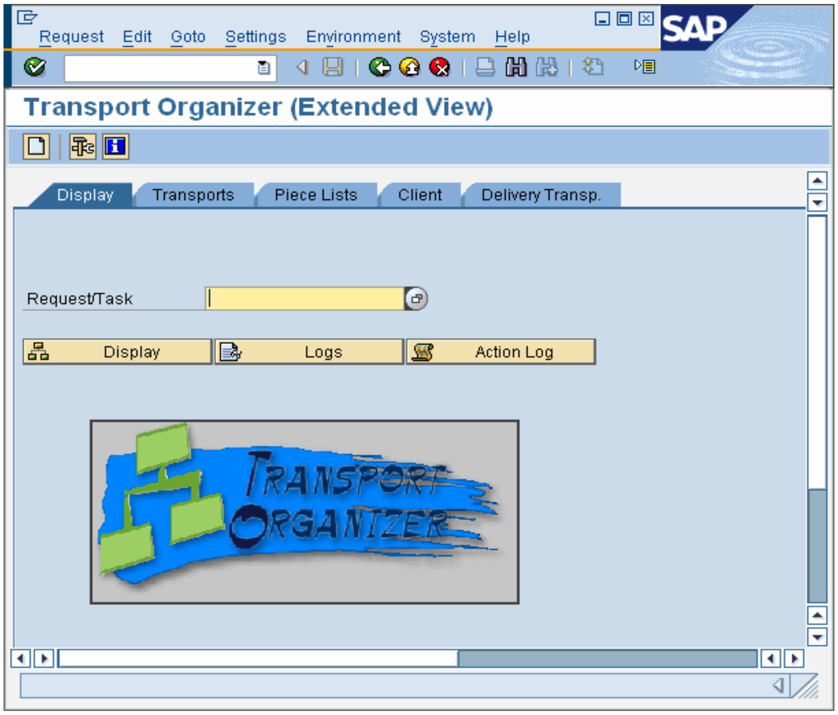
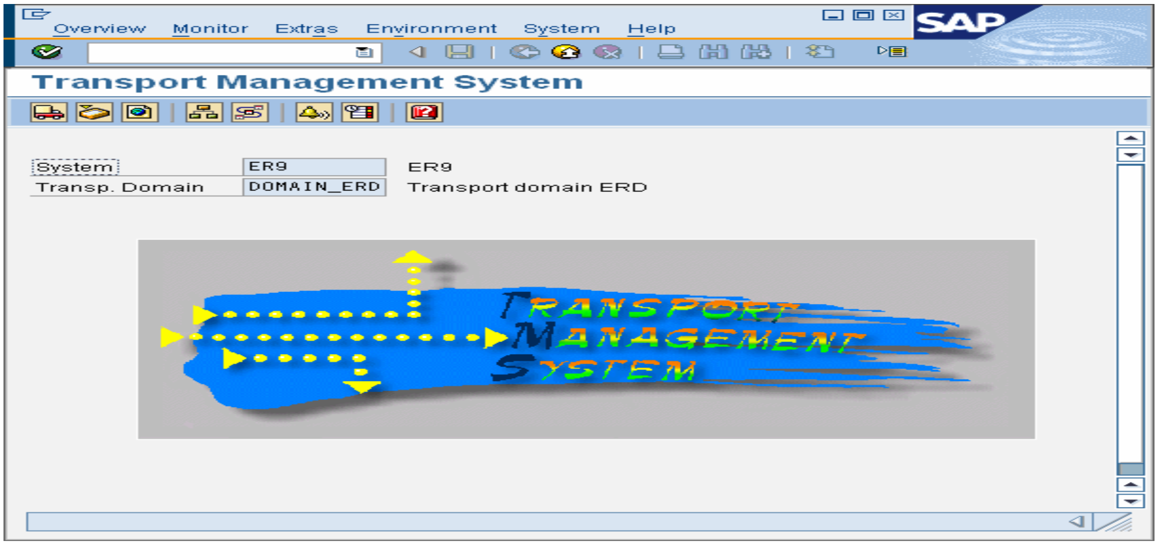


Fig. 5. Transport organizer.

- Transport Management System (TMS) [4], [5]: It mainly distributes changes recorded by the CTO across SAP systems in a systematic way. The transport control program, ‘tp’ executes these changes at the operating system level. Transaction STMS, shown in Figure 6, is used to configure and administer transports.



Transport Management System

System ER9 ER9
Transp. Domain DOMAIN_ERD Transport domain ERD

Fig. 6. Transport management system.

Changes are made in the development system. These changes are then moved over to a Quality Assurance (QA) system for testing those changes. After successful tests, the changes are then moved into Production. This modification and transfer is accomplished by the Change and Transport System or the CTS.

The CTS is one of the most important components. The CTS should be configured before developers can use the system for development otherwise none of the changes can be tracked or moved to other systems. All standard customizations are carried out in IMG (Implementation Guide) and developments in ABAP workbench. The Change and Transport System records all the changes to the Customizing and Repository data. If CTS is not configured correctly, the following situations may occur which will result in unexpected results as follows:

- Changes made by developers will not prompt for transport creation. This will cause loss of integrity of objects. For example, when developer creates a table, many underlying “basis” tables hold the data. If the system does not capture all of these objects into a change request, it can never be moved to the Quality Assurance (QA) system for testing.
- If the transport routes are not set correctly, the transports (change request) released from development system will never be delivered to the Production system.

The screenshot in Figure 7 depicts the transport routes. The system ERD (System Id) is the development system that holds all the customizations and developments. The transport layers SAP and ZERD are the consolidation routes to move standard SAP and custom changes respectively into QA system. The delivery route then moves the changes over to the targets. There can be more than one consolidation and delivery systems. Also, the consolidation and delivery systems need not be an actual system. It can be a virtual system which will hold all the changes to be moved to the targets.

The following describes the main entities while setting up Transport Management System.

- Transport Domain Controller [4], [5]: The system that holds all the systems’ configuration information.
- Transport Route [4], [5]: The transport routes define which targets to consolidate change requests.

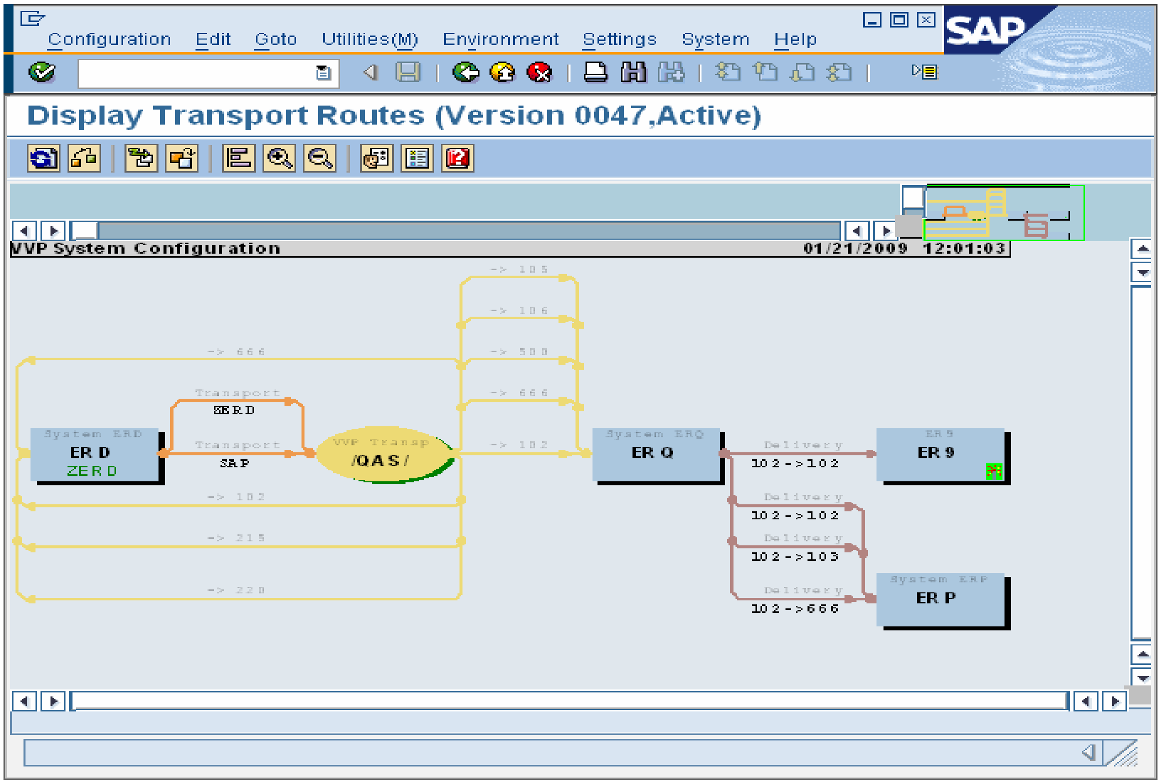


Fig. 7. SAP transport routes.

## SAPconnect

All communication external to SAP system is carried out by SAPconnect [5]. SAPconnect supports sending of emails to Internet addresses, faxes, test messages etc. SAPconnect uses SAP Netweaver’s SMTP plug-in for this purpose. It can also leverage the services from any external communication system. SAPconnect administration is done through transaction SCOT.

### <u>Configuration E-mail, Fax, Paging/SMS via SMTP</u>

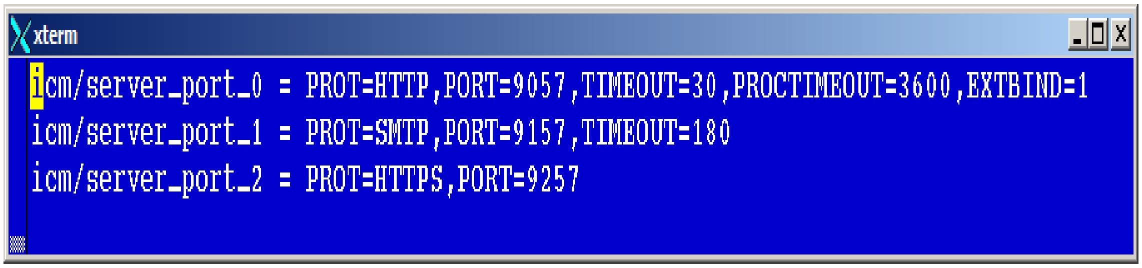
SMTP should be activated in the SAP system. In addition to this, a standard job should be scheduled to send emails out of the system. Without performing the correct configuration, emails, pages will never be sent out of the SAP system. System monitoring functions send pages from the system to the administrators hence SMTP configuration is very critical.

The profile of the SAP Web Application Server must be adjusted to use SMTP functions.

- icm/server_port_<*> =   PROT=SMTP   PORT=<port>

TCP/IP port is opened for the receipt of mails via the SMTP plug-in. PORT specifies the port number to be used. No other program should occupy this port on this host. PORT can be set to 0 if no emails are to be received.

Figure 8 shows the profile parameters that need to set in order to enable SMTP:

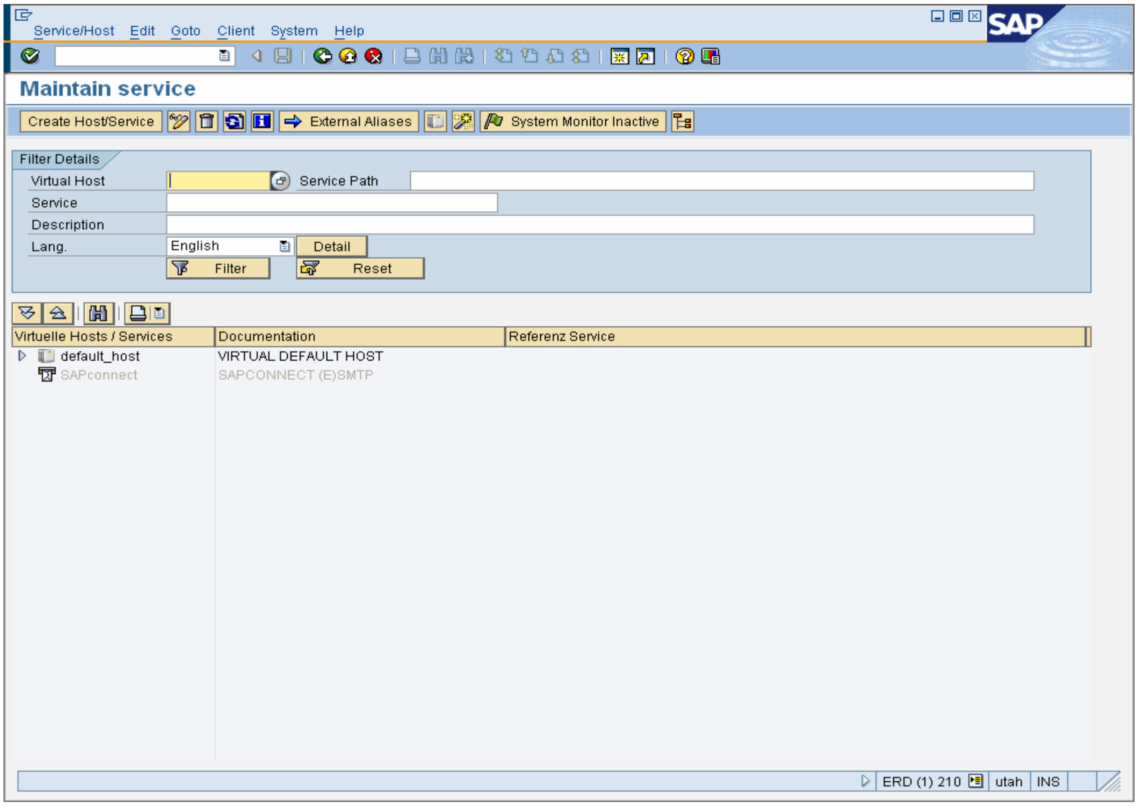


```
icm/server_port_0 = PROT=HTTP,PORT=9057,TIMEOUT=30,PROCTIMEOUT=3600,EXTBIND=1
icm/server_port_1 = PROT=SMTP,PORT=9157,TIMEOUT=180
icm/server_port_2 = PROT=HTTPS,PORT=9257
```

Fig. 8. ICM profile parameters.

## SMTP Plug-in Activation (Transaction SICF)

An SMTP server must be created on each client where incoming mails are intended to be received and processed. In transaction SICF, an SMTP plug-in should already be available in every SAP system. This is delivered by SAP. This plug-in should be activated in order to have SMTP activated. This particular plug-in resides under the default node of all the ICF services. The SMTP server must be activated after being created or changed (Service/Virt. Host → Activate). Figure 9 shows that the node by default is disabled.



| | |
|---|---|
| Virtual Host | |
| Service Path | |
| Service | |
| Description | |
| Lang. | English |

| Virtuelle Hosts / Services | Documentation | Referenz Service |
|---|---|---|
| default_host | VIRTUAL DEFAULT HOST | |
| SAPconnect | SAPCONNECT (E)SMTP | |

ERD (1) 210 | utah | INS

Fig. 9. Maintain ICF services.

## SAPconnect Administration (Transaction SCOT)

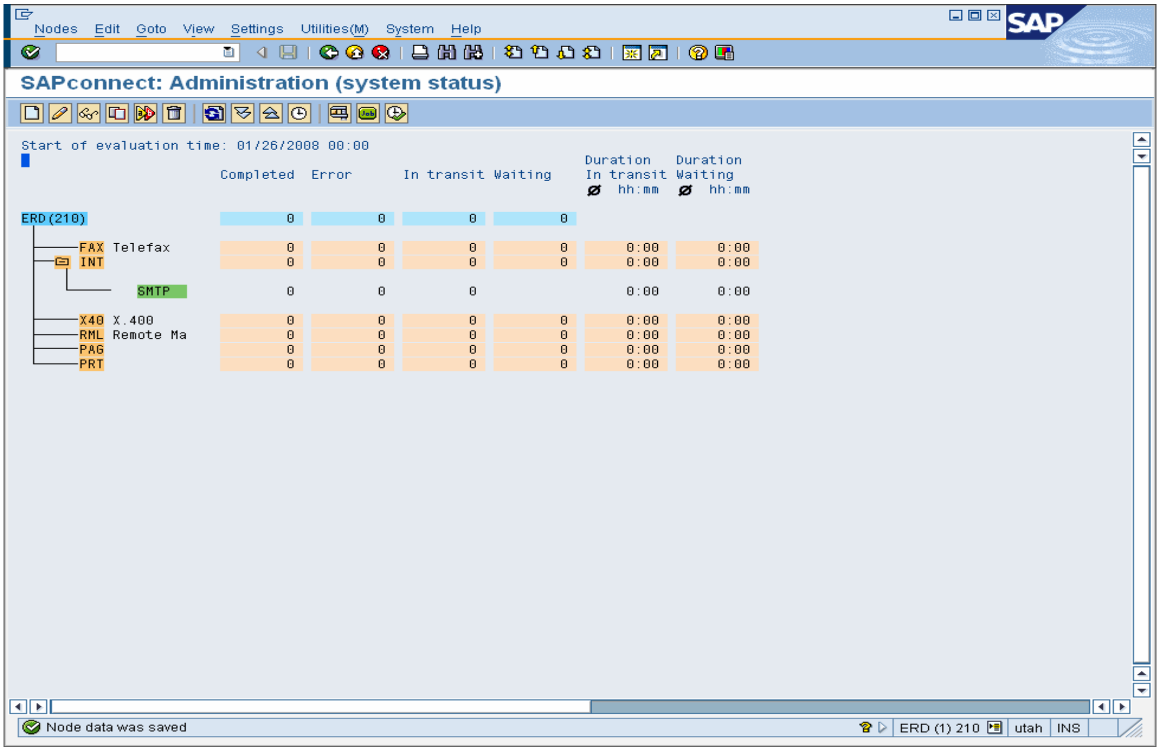
SAPconnect settings are separately made in each client from which e-mails are to be sent or in which e-mails are to be received.

a) Default domain is the default domain of the SAP system client is defined here (e.g., utah.varian.com). It is used for the following purposes:

- The SMTP plug-in uses domain as ID to log on to the mail server.
- The message ID of outgoing mails.

- Generate an email address if an SAP user forgets to specify an Internet address, for example, Arvind.Nair@varian.com.

b) Node. There are different types of nodes in SAPconnect as shown in Figure10.



SAPconnect: Administration (system status)

Start of evaluation time: 01/26/2008 00:00

| | | Completed | Error | In transit | Waiting | Duration<br>In transit<br>Ø hh:mm | Duration<br>Waiting<br>Ø hh:mm |
|---|---|---|---|---|---|---|---|
| ERD(210) | | 0 | 0 | 0 | 0 | | |
| FAX | Telefax | 0 | 0 | 0 | 0 | 0:00 | 0:00 |
| INT | | 0 | 0 | 0 | 0 | 0:00 | 0:00 |
| SMTP | | 0 | 0 | 0 | | 0:00 | 0:00 |
| X40 | X.400 | 0 | 0 | 0 | 0 | 0:00 | 0:00 |
| RML | Remote Ma | 0 | 0 | 0 | 0 | 0:00 | 0:00 |
| PAG | | 0 | 0 | 0 | 0 | 0:00 | 0:00 |
| PRT | | 0 | 0 | 0 | 0 | 0:00 | 0:00 |

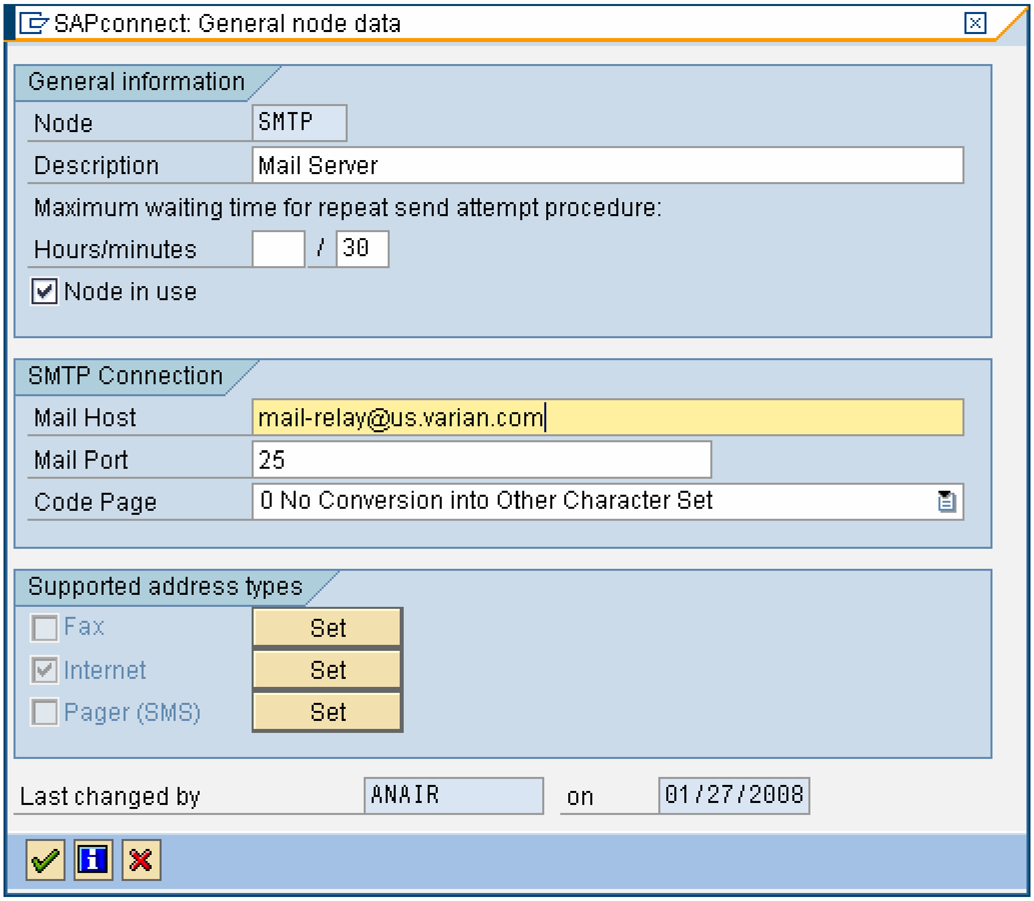
Node data was saved

ERD (1) 210 | utah | INS

Fig. 10. SAPconnect nodes.

- SMTP node (for the SMTP function of the SAP application server)
- HTTP node (for paging/SMS providers using Web services)
- RFC node (for old RFC-compatible e-mail/fax/paging gateways)

In each client has a single SMTP node. It cannot be deleted as it is automatically created. It is configured as follows: Setting the mail host correctly, as shown in Figure 11, is vital to SAPconnect's functioning. The destination host is the gateway to the external world. All emails, faxes, SMS are sent to mail host for delivery.



SAPconnect: General node data

General information

Node: SMTP

Description: Mail Server

Maximum waiting time for repeat send attempt procedure:

Hours/minutes: / 30

☑ Node in use

SMTP Connection

Mail Host: mail-relay@us.varian.com

Mail Port: 25

Code Page: 0 No Conversion into Other Character Set

Supported address types

□ Fax Set

☑ Internet Set

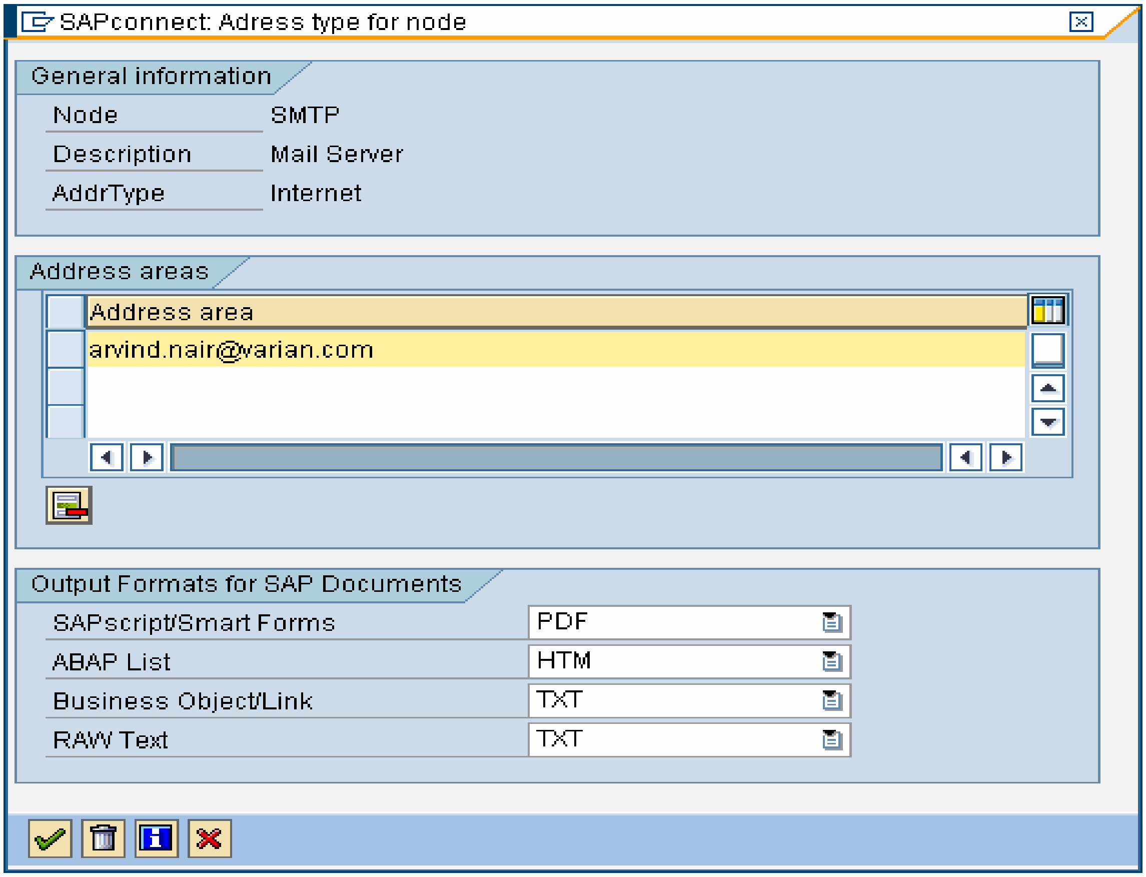
□ Pager (SMS) Set

Last changed by ANAIR on 01/27/2008

Fig. 11. Setting up mailhost information in SAPConnect.

Additionally email address, faxes should be set up in SAP and it controls all the messages that can be sent out of SAP to the mail host for delivery. It much be ensured that 'Node in use' checkbox must always be checked. The recipient list can be added in the any of the address types. For emails, the 'Internet' checkbox needs to be checked and all the email addresses can be entered but click on that 'Set' button against it which is shown in Figure 12.

E-mails sent from an SAP are queued. A job is scheduled. This job sends all accumulated emails to the recipients mentioned in the email addresses. Following are the steps to schedule the job:



SAPconnect: Adress type for node

General information

| | |
|---|---|
| Node | SMTP |
| Description | Mail Server |
| AddrType | Internet |

Address areas

| Address area |
|---|
| arvind.nair@varian.com |

Output Formats for SAP Documents

| | |
|---|---|
| SAPscript/Smart Forms | PDF |
| ABAP List | HTM |
| Business Object/Link | TXT |
| RAW Text | TXT |

Fig. 12. Email addresses and output formats.

1. In transaction SCOT, from the menu choose View -> Jobs.

2. Choose Job → Create

3. Choose Schedule Job

4. Choose periodicity to be 10 minutes.

# CHAPTER IV

# DESIGN

In real business world, there are continual changes. There are changes in business processes and department functions. The changes are made by either enhancing a certain module or by implementing new modules. For example, the HR department wants employees to maintain their personal information, emergency contact information instead of the manual maintenance by an HR administrator. There are innumerable such changes, in the form of enhancement to current process, new processes that happens in each department of a company. The changes could be development, configuration change or a system change. All the changes have to be documented with appropriated approvals, testing and validation.

All the changes in the system are made in development and then moved to the quality assurance (QA) system after unit tests. In the QA system, the integration test and user acceptance test (UAT) are carried out before moving it to production. The changes are then reviewed by the change review board (CRB), a committee of analysts, both functional and technical. After the approval from CRB, the set of changes are then moved to production system. These changes are reviewed by the internal auditors quarterly and the external auditors, annually.

To keep track of all these changes, approvals, tests and validation results is huge and there is lot of effort to keep track of the email chains of approvals. Hence, the SAP change management tool provides the right solution to document the changes happening across SAP landscape with appropriate compliance to Sarbanes-Oxley (SOX). Auditors look for proper process around change management with appropriate requests, resolution, testing, validation and review of all the changes that are going to Production.

The SAP change management tool is used by developers, analysts, business users and business owners, CRB members and auditors. All the interactions with the tool can be depicted in Figure 13. Each actor interacts with the use cases as below:

- Developers:
  - Create change request for
    - Bug fixes
    - New developments or configurations
  - Add transports to the change request
  - Request the transports to be moved from development to QA and from QA to Production review.
  - Update code review documents, unit test and integration test documents, SOX implications.
- Analyst:
  - Review the development/configuration changes
  - Approve the transport to QA
  - Validate all documents and modify if required

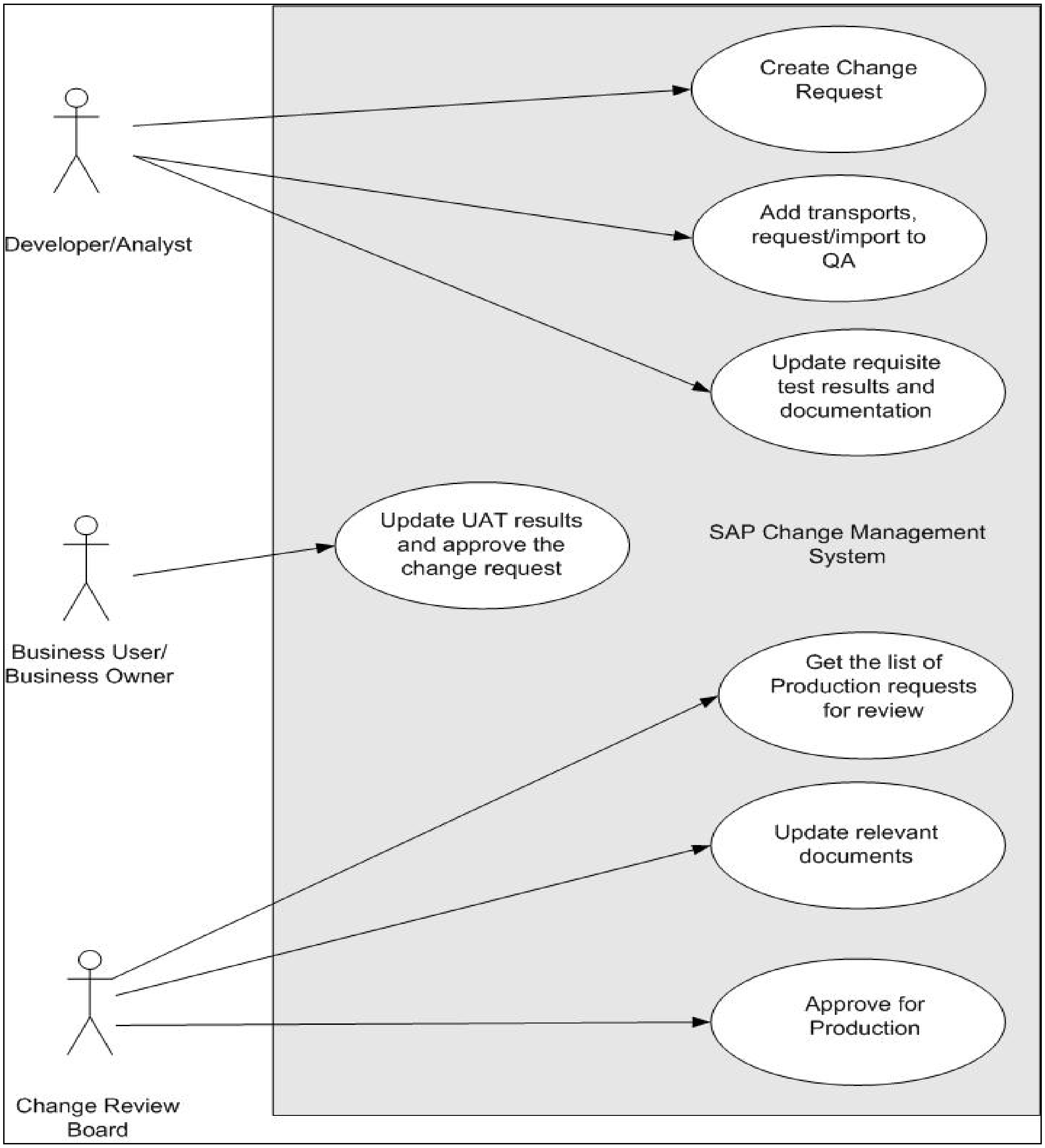


Fig. 13. Use case diagram.

- Business User:
  - Update UAT results
  - Approve the change request.

- Business Owner
    - Approve the change request
- Change Review Board (CRB):
    - Review all the changes and update relevant approval documents.
    - Approve the change request to Production.

The use case scenario explained above can be detailed by means of the sequence diagram shown in Figure 14.

The sequence diagram in Figure 14 can be explained as follows:

- The developer opens change request for any bug that is fixed or a project. The change is made in development and once the change is unit tested, the developer attaches the transport to the change requested.
- The analyst approves the development changes to QA.
- The business user performs user acceptance test and approves the change request.
- The business owner then approves the changes to be moved to production
- All the changes that need to go to Production are reviewed by the CRB. The CRB then approves the transport to Production.

The use diagram and sequence diagram explains in detail about the underlying process that takes place at each state of a change request. The status of the change request can be maintained on any of the following depending on what activities have been done for that particular change request. It is important to maintain these states to give an idea to the management about the current statuses of all the changes that are happening across the SAP landscape.

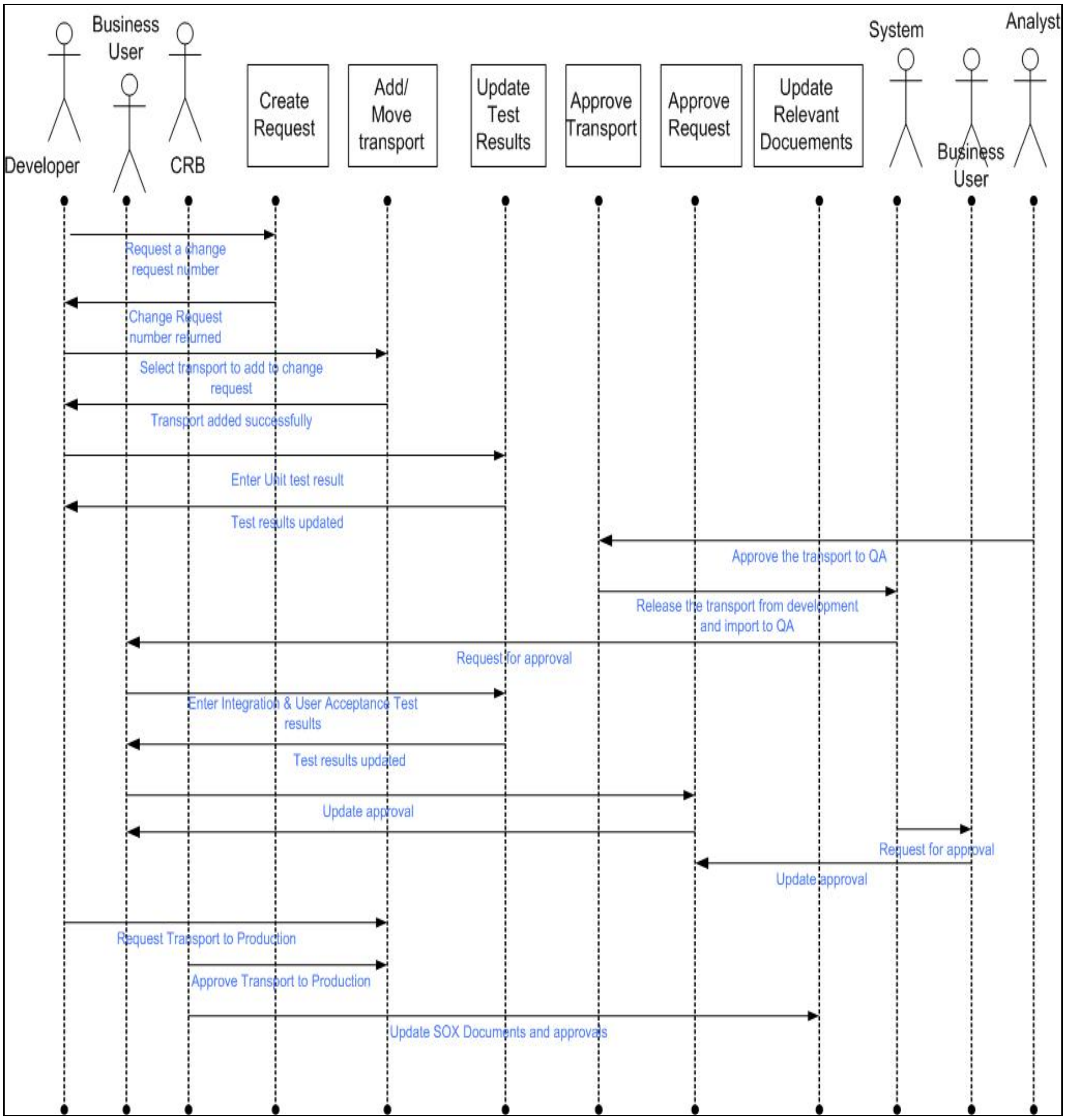

Developer
Business User
CRB
Create Request
Add/ Move transport
Update Test Results
Approve Transport
Approve Request
Update Relevant Docuements
System
Business User
Analyst
Request a change request number
Change Request number returned
Select transport to add to change request
Transport added successfully
Enter Unit test result
Test results updated
Approve the transport to QA
Release the transport from development and import to QA
Request for approval
Enter Integration & User Acceptance Test results
Test results updated
Update approval
Request for approval
Update approval
Request Transport to Production
Approve Transport to Production
Update SOX Documents and approvals

Fig. 14. Sequence diagram.

- Created: This is the initial state of change request upon inception.
- In Development: Once a developer or the person making any change starts working on the change request, then the state of the request is changed to “In Development”

- Requirement Gathering: In case of projects, there will be times when developer needs to get requirements from end user and hence can change this status accordingly.
- Assessing LOE: In cases where the projects or the defects are very complex, there might be a need to park the change request in “Accessing Level of Effort” status.
- In Unit Test: When developer tests the changes in the development after development is done, the status can be changed to this status.
- In Integration Test: The change request has this status when integration test is performed in the QA system.
- In User Acceptance Test: This status is maintained when the end user is testing the defect or functionality in QA system.
- Signed Off: When business user approves the request, this status is maintained in the change request.
- Closed/In Production: When the transports are moved to Production, then this status can be maintained.
- Rejected: When the changes are not approved by the developer if the change request need not be sent to Production for some reason.
- On Hold: This status may be maintained if the change request has some dependencies on other change request or it should be sent to Production at a later point in time.

The states of the transports attached to the change request are as follows:

- In development: A change request is in development state when the transports attached to it are in the development system awaiting release to QA.
- Requested for QA: A change request is in “Requested to QA” state when a developer requests the transports to be released to QA after completing unit test in the development system.
- QA approved: A change request is in this state when an analyst has verified that the transports are ready to be moved to QA and releases the transports from the development system.
- Production Requested: After the changes are tested in the QA system by the requestor and approved by business owner, the developer then requests the transports to be released to Production
- Production Approved: The CRB lists all the transports to be approved for Production and then after reviewing each change then approves the changes to Production.

The complete scenario that has been explained with the help of sequence diagram can be depicted by means of activity diagram in Figure 15.

Each process in the sequence diagram and the activity diagram can be mapped to a function. The functions form the base functionality of the application. Hence there will be following functions:

- Create/Change/Display Request
- Add transport

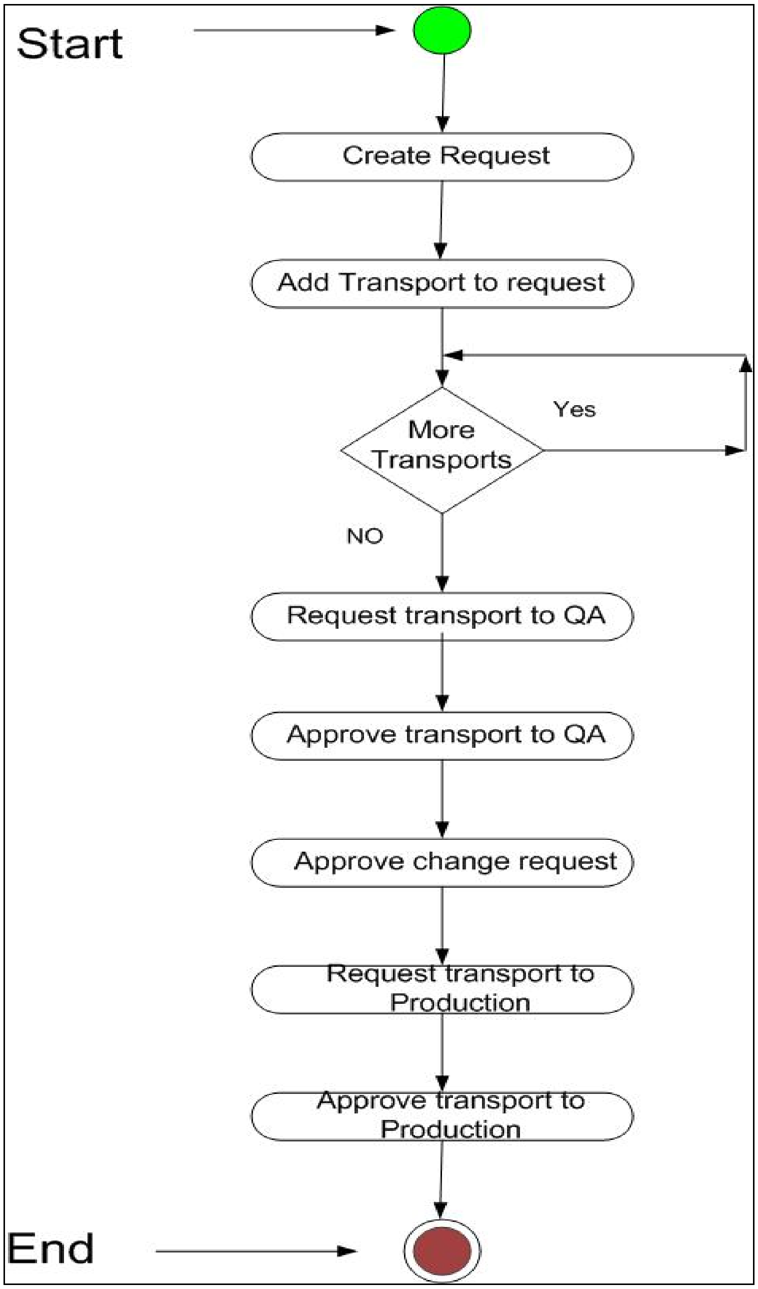


Fig. 15. Activity diagram.

- Update document/test results/Code review
- Release transports
- Approve transport to QA/Production

- Approve/Reject Request
- Email notification

The following reports also have to be developed:

- Management Report of the change requests: Give management or auditors an overview of the change requests in the system
- Transports in the change request: This report gives the transports attached to the change request
- Transport objects in the change request: The report gives all the objects that will go to production if the change request is moved to Production.
- Search Change Requests.

The system should also have pockets to store the following documentations:

- Sarbanes-Oxley (SOX) documents
- Test Documents
    - Unit test
    - Integration test
    - User Acceptance test
- Technical review in case there are development changes.
- Training documents in case of new functionality or projects
- Business Justification documents
- Justification for change request and Solution provided for the issues being addressed.

# CHAPTER V

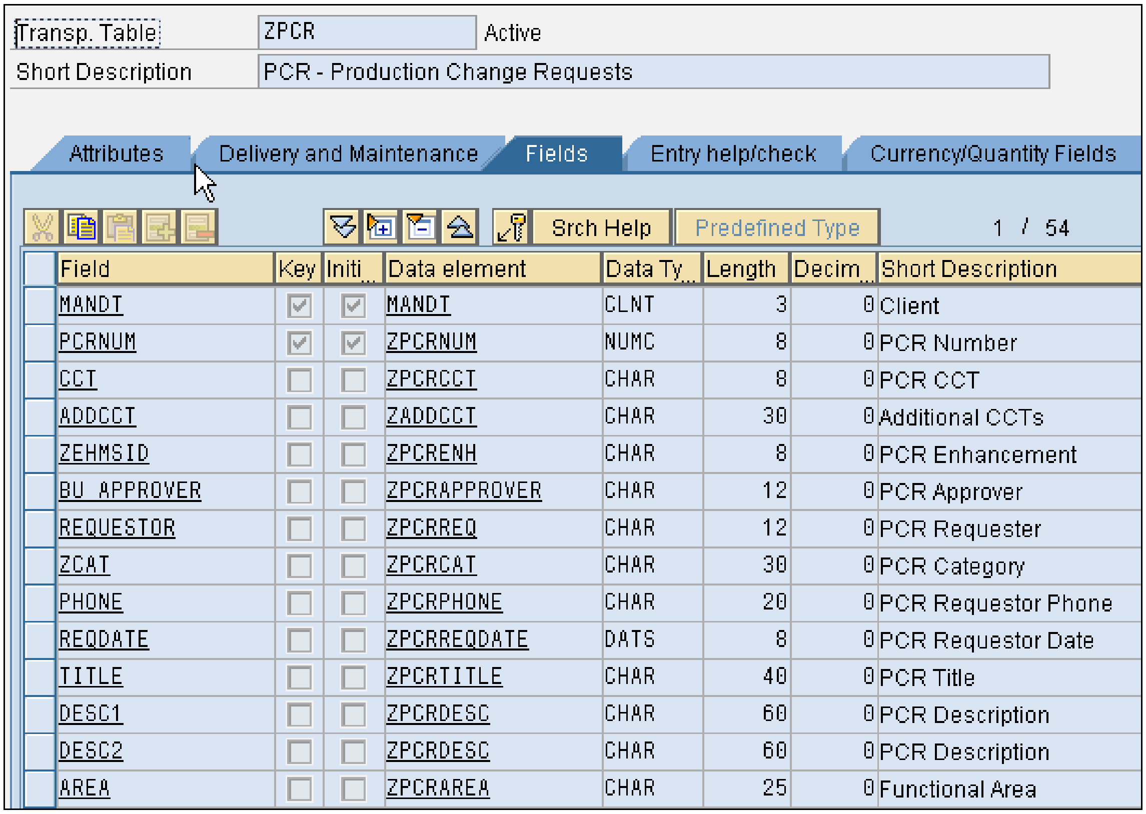
# IMPLEMENTATION

The change management system has been built over SAP’s Transport Management System. Many components make the change management system. The following are the elements that have gone into the building blocks of the tool:

- Tables
- Domains and Data Elements
- Structures and Internal tables
- Programs and Reports
- Screens
- Function Modules
- Transactions

## Building Blocks of the Application

The tables used in the application holds master data and transactional data. The master data tables hold functional area, type of change request, requestor and the systems. The transactional data tables hold the change request itself, the test history, the application log, the code review and check results. The main table of the application is shown in Figure 16 that holds all the change requests.



Transp. Table ZPCR Active

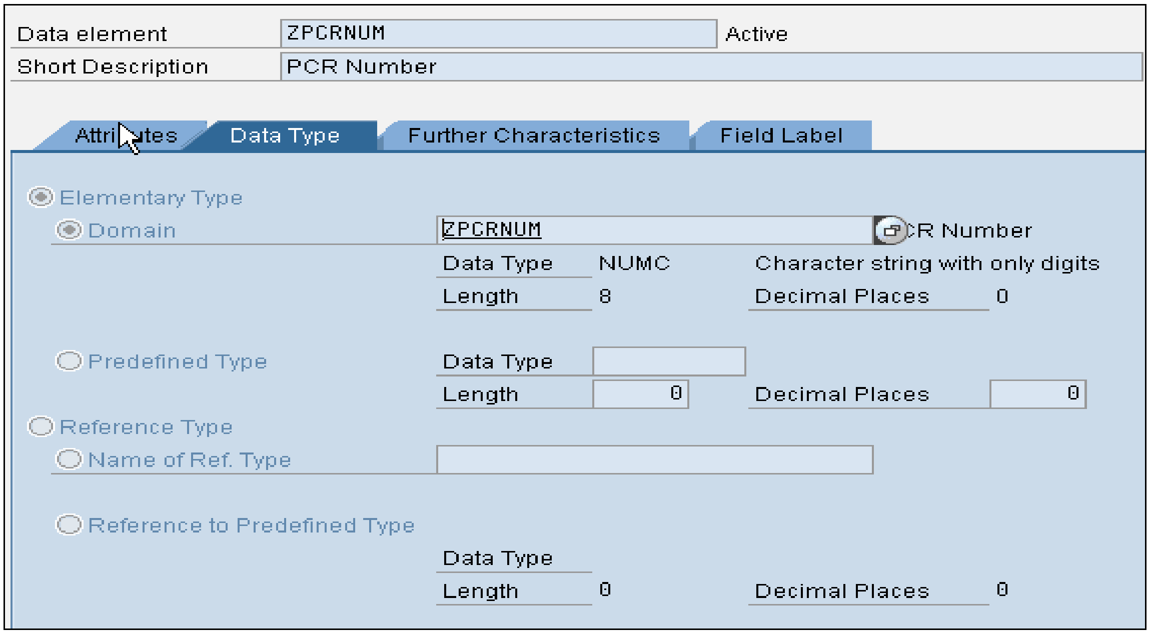
Short Description PCR - Production Change Requests

| Field | Key | Initi... | Data element | Data Ty... | Length | Decim... | Short Description |
|---|---|---|---|---|---|---|---|
| MANDT | ☑ | ☑ | MANDT | CLNT | 3 | 0 | Client |
| PCRNUM | ☑ | ☑ | ZPCRNUM | NUMC | 8 | 0 | PCR Number |
| CCT | □ | □ | ZPCRCCT | CHAR | 8 | 0 | PCR CCT |
| ADDCCT | □ | □ | ZADDCCT | CHAR | 30 | 0 | Additional CCTs |
| ZEHMSID | □ | □ | ZPCRENH | CHAR | 8 | 0 | PCR Enhancement |
| BU_APPROVER | □ | □ | ZPCRAPPROVER | CHAR | 12 | 0 | PCR Approver |
| REQUESTOR | □ | □ | ZPCRREQ | CHAR | 12 | 0 | PCR Requester |
| ZCAT | □ | □ | ZPCRCAT | CHAR | 30 | 0 | PCR Category |
| PHONE | □ | □ | ZPCRPHONE | CHAR | 20 | 0 | PCR Requestor Phone |
| REQDATE | □ | □ | ZPCRREQDATE | DATS | 8 | 0 | PCR Requestor Date |
| TITLE | □ | □ | ZPCRTITLE | CHAR | 40 | 0 | PCR Title |
| DESC1 | □ | □ | ZPCRDESC | CHAR | 60 | 0 | PCR Description |
| DESC2 | □ | □ | ZPCRDESC | CHAR | 60 | 0 | PCR Description |
| AREA | □ | □ | ZPCRAREA | CHAR | 25 | 0 | Functional Area |

Fig. 16. Display table columns.

The fields of the tables are defined by data types that represent the type of value it stores. The data types can either be a standard data type or can be user defined data type. The data elements in the above screenshot are created because there is no standard data element that can be used to qualify the table fields. The data elements themselves are defined as a predefined data type, a domain type or reference data type. In the Figure 17, the data element ‘ZPCRNUM’ is defined as a domain type ‘ZPCRNUM’.

The domain ‘ZPCRNUM’ shown in Figure 18 is defined as a numeric data type of length 8. It is used to create data types from standard data types for qualifying data elements.



Data element ZPCRNUM Active

Short Description PCR Number

● Elementary Type

● Domain ZPCRNUM CR Number

Data Type NUMC Character string with only digits

Length 8 Decimal Places 0

○ Predefined Type Data Type

Length 0 Decimal Places 0

○ Reference Type

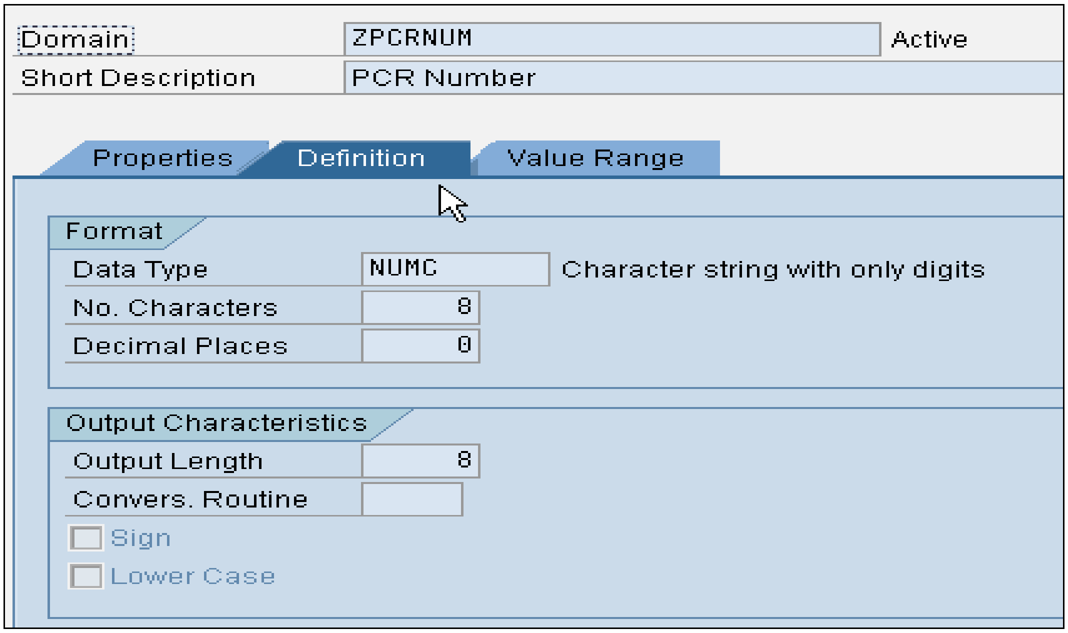
○ Name of Ref. Type

○ Reference to Predefined Type

Data Type

Length 0 Decimal Places 0

Fig. 17. Data element.



Domain ZPCRNUM Active

Short Description PCR Number

Format

Data Type NUMC Character string with only digits

No. Characters 8

Decimal Places 0

Output Characteristics

Output Length 8

Convers. Routine

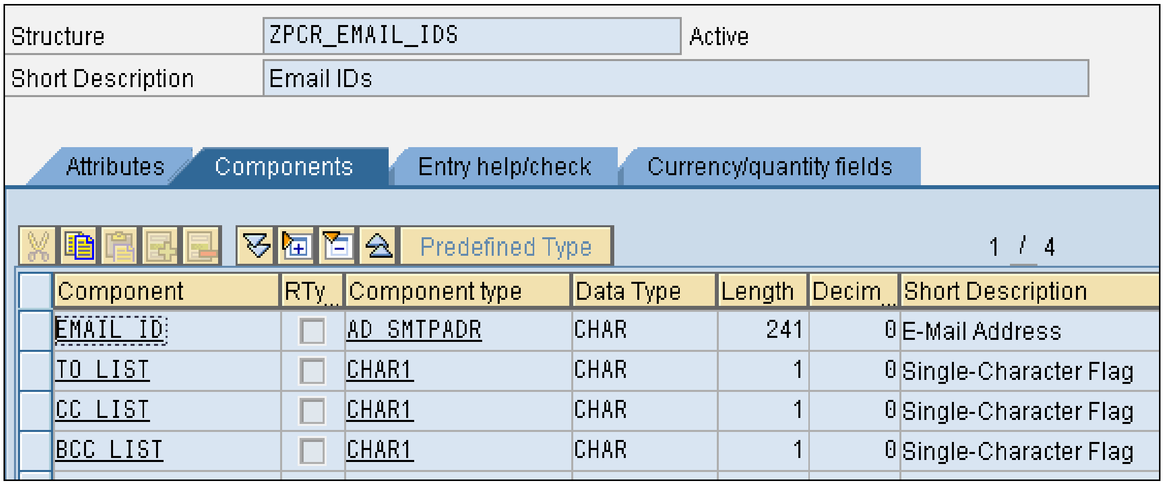
□ Sign

□ Lower Case

Fig. 18. Domain.

The structure in ABAP is a data type that is filled during the runtime of the program. It is defined as group of data elements bundled together. Structure can either have elementary data elements making its composition or can be built of complex structures with reference types, nested structures and internal tables. Internal tables are ABAP structures that are defined as table types. Internal tables themselves can contain other internal tables.

The screenshot Figure 19 shows a simple structure that is composed of elementary data types. The data in structure have no data permanence but are filled during runtime and cleared after each run. They are held in memory for processing when the program or report using it is loaded into the memory. The internal tables too are similar to structure data types.


Structure: ZPCR_EMAIL_IDS Active
Short Description: Email IDs

| Component | RTy... | Component type | Data Type | Length | Decim... | Short Description |
|---|---|---|---|---|---|---|
| EMAIL_ID | □ | AD_SMTPADR | CHAR | 241 | 0 | E-Mail Address |
| TO_LIST | □ | CHAR1 | CHAR | 1 | 0 | Single-Character Flag |
| CC_LIST | □ | CHAR1 | CHAR | 1 | 0 | Single-Character Flag |
| BCC_LIST | □ | CHAR1 | CHAR | 1 | 0 | Single-Character Flag |

Fig. 19. Structure.

Programs and reports are callable units that have processing blocks that are called in particular sequence. The processing blocks consist of a number of structures and subroutines that are called in a predefined sequence. Programs can either be run directly or can be called by other executable units. Programs that are directly executable are knows as reports and can be called by a transaction or the program name itself. Programs can also call screens which consist of dialog modules that control the program flow. There are processing units before and after screen output. Screen can also be linked to transactions such that the program associated to the transaction calls the screen first in the list of processing units. The graphical screen painter shown below in Figure 20 is used to create screens.

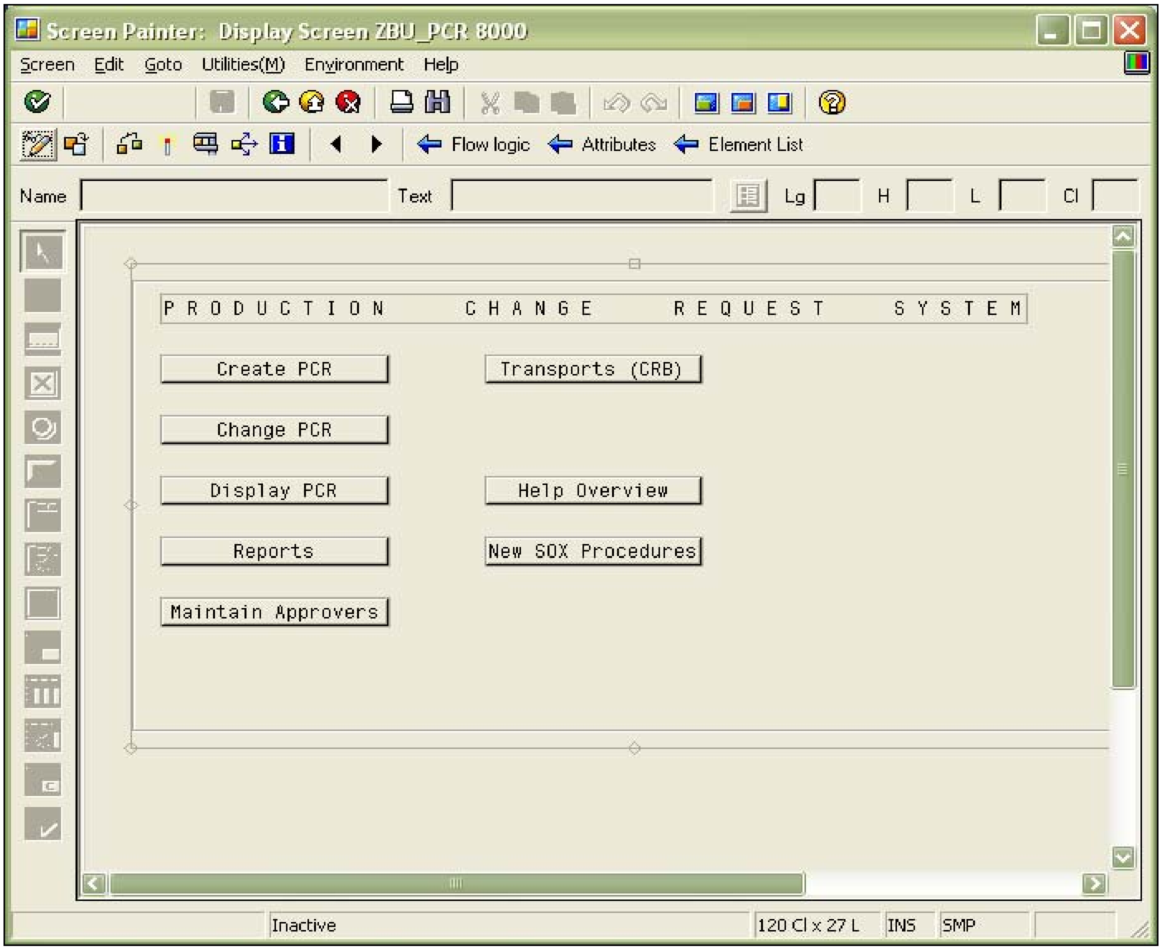


Fig. 20. Screen for program in graphical screen painter.

Function Modules on the other hand are executable units that are defined for specific tasks that are called single or multiple times during program execution. Function modules may take a set of import parameters and may return a set of export parameters to the calling program.

## PCR Application Schema

Figure 21 shows the main schema of the application. The table PCR holds all the change requests in the system. The field PCR_ID being the primary key identified each change requests uniquely. The table TEXT holds all the text that is entered for each PCR. The text for change requests are:

- Justification
- BW Impact
- General Description
- Security Impact
- SOX Impact
- Impact Analysis
- Test Info/results
- Solution Log
- Recovery Procedure

The change request change log is updated each time any change is made to any change request in the system. All the transports or changes are updated in the transport request table. The Solutions table holds the resolution of the issue.

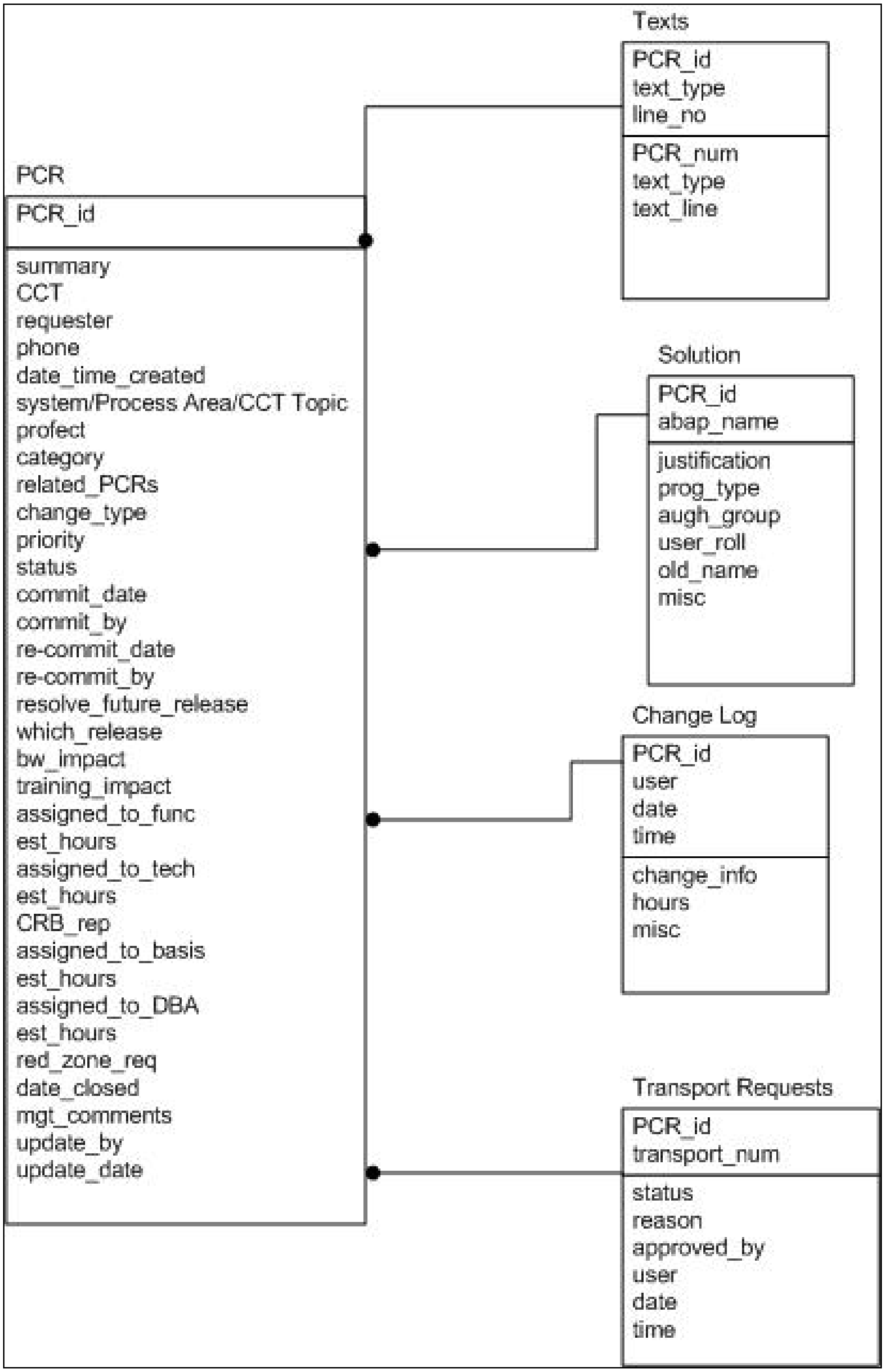


Fig. 21. PCR schema.

## Application Walkthrough

The application is started by transaction ZPCR. This transaction runs the associated program and its processing units. The screen that is associated with the program runs the flow logic is output to the GUI. The respective application memory structures are cleared. The process before output (PBO) flow logic runs before the screen is called. The main screen of the application is shown in Figure 22.

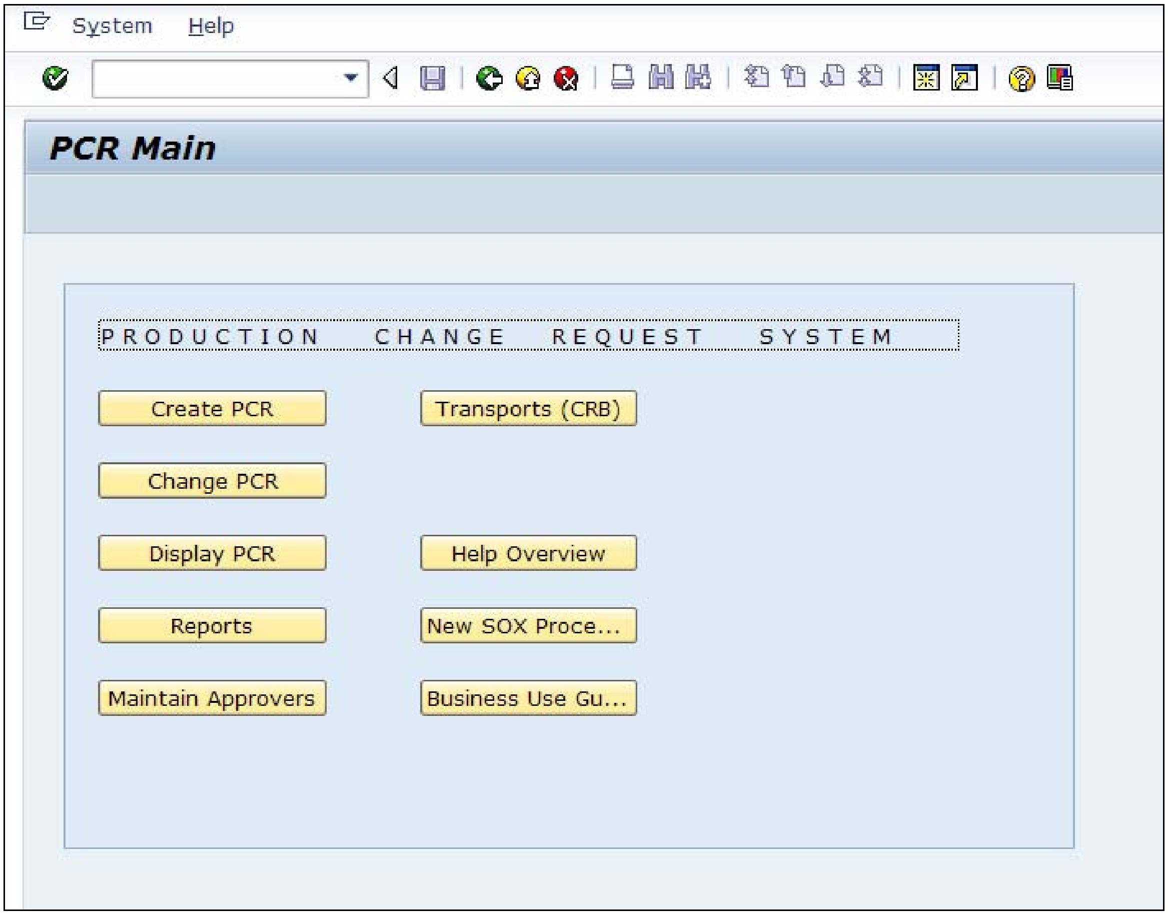
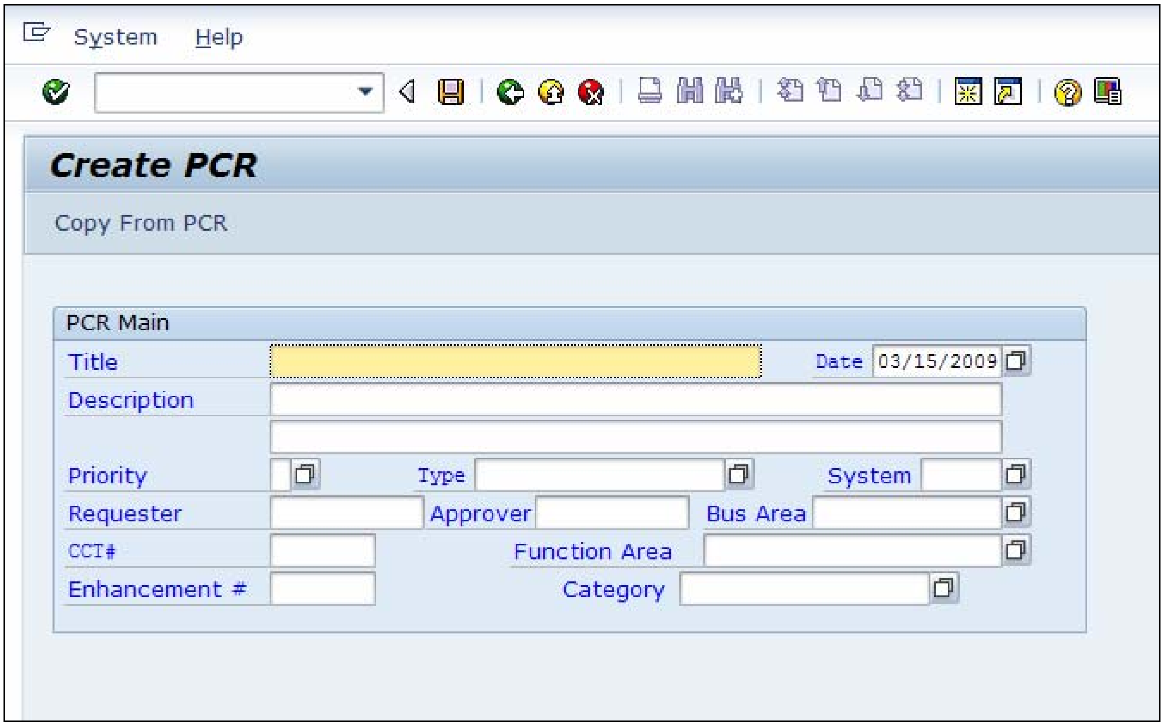


Fig. 22. PCR main application.

After the transaction is called, all internal structures are initialized. All the push buttons in the above figure runs events that trigger further actions. According to the figure there are following actions depending on which stage the change request is.

- Create PCR: For all new change requests that need to be entered into the system.
- Change PCR: To make changes to any existing change requests.
- Display PCR: To view change request.

Once the push button "Create PCR" is clicked, the screen shown in Figure 23 appears asking the developer to input change request information.



Create PCR

PCR Main

Title

Date 03/15/2009

Description

Priority

Type

System

Requester

Approver

Bus Area

CCI#

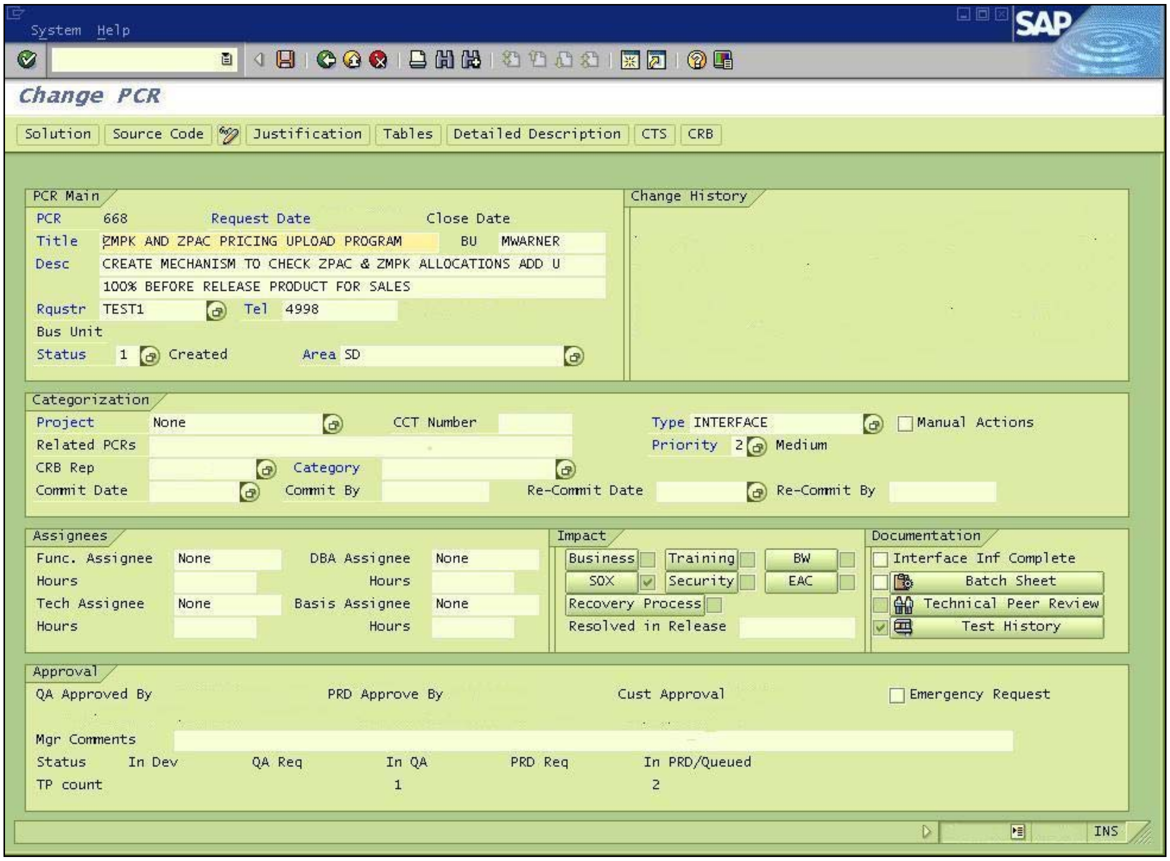
Function Area

Enhancement #

Category

Fig. 23. Create production change requests.

The title and the description of the change request are filled appropriately and they give clarity to the change request. The 'Priority' of the change request determines how it should be processed by the change review board. The ‘Type’ of the change request indicates if the change is a development change, configuration change or system change. The ‘System’ field indicates the system in which the change is made. ‘Bus Area’ indicates the business area in which the change is made. ‘Approver’ is important since business owner should approve change. If the change request is an enhancement then that is filled in the ‘Enhancement#’ field. All the fields give audit enough information about the change. After the save button is clicked, a sequence number is generated and the change request number is created. In Figure 24, the details of the change request are filled.



Change PCR

PCR Main

PCR 668 Request Date Close Date

Title ZMPK AND ZPAC PRICING UPLOAD PROGRAM BU MWARNER

Desc CREATE MECHANISM TO CHECK ZPAC & ZMPK ALLOCATIONS ADD U
100% BEFORE RELEASE PRODUCT FOR SALES

Rqustr TEST1 Tel 4998

Bus Unit

Status 1 Created Area SD

Change History

Categorization

Project None CCT Number Type INTERFACE □ Manual Actions

Related PCRs Priority 2 Medium

CRB Rep Category

Commit Date Commit By Re-Commit Date Re-Commit By

Assignees

Func. Assignee None DBA Assignee None

Hours Hours

Tech Assignee None Basis Assignee None

Hours Hours

Impact

Business □ Training □ BW □

SOX ☑ Security □ EAC □

Recovery Process □

Resolved in Release

Documentation

□ Interface Inf Complete

□ Batch Sheet

□ Technical Peer Review

☑ Test History

Approval

QA Approved By PRD Approve By Cust Approval □ Emergency Request

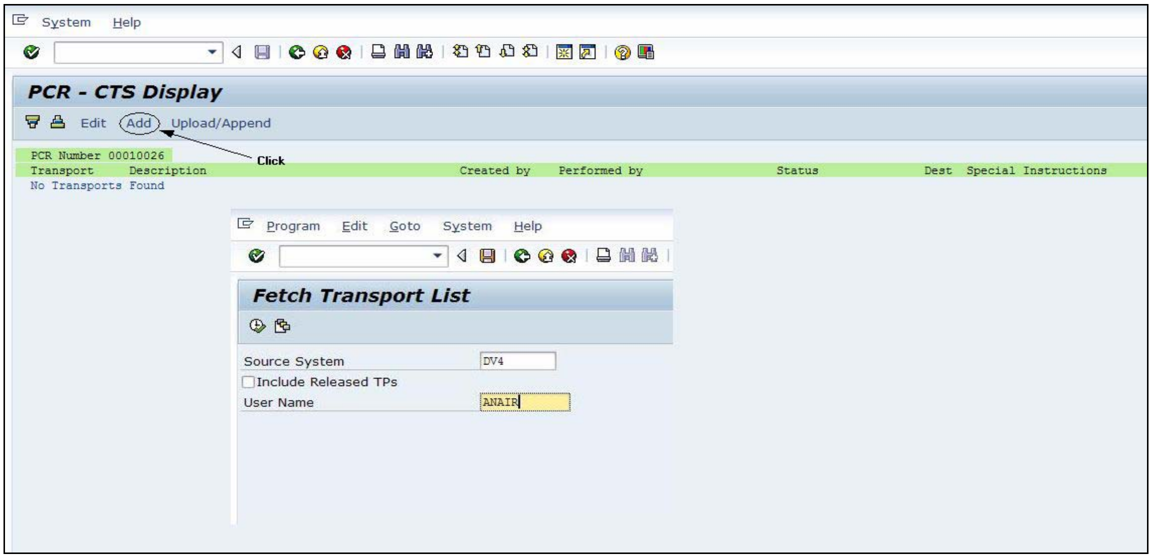
Mgr Comments

Status In Dev QA Req In QA PRD Req In PRD/Queued

TP count 1 2

Fig. 24. Change PCR.

The details of the change request are stored in the table PCR. Now after the developer completes the development tasks and then updates the change request with the transport number. In order to add the transport number to the change request, the push button ‘CTS’ has to clicked which will then call another screen which is shown below. The screen will initially show no transports. Now after clicking the ‘Add’ sub menu, a dialog window pops up and requests for the transport request as shown in Figure 25. The system in which the transport has been created is selected. A function module then calls the system to query for the transports for the user. The listed transport can then be added to the change request and since the transport is in development system, the status of the transport of the change request reflects the status appropriately.



PCR Number 00010026

```
Transport     Description                              Created by   Performed by          Status          Dest  Special Instructions
No Transports Found
```

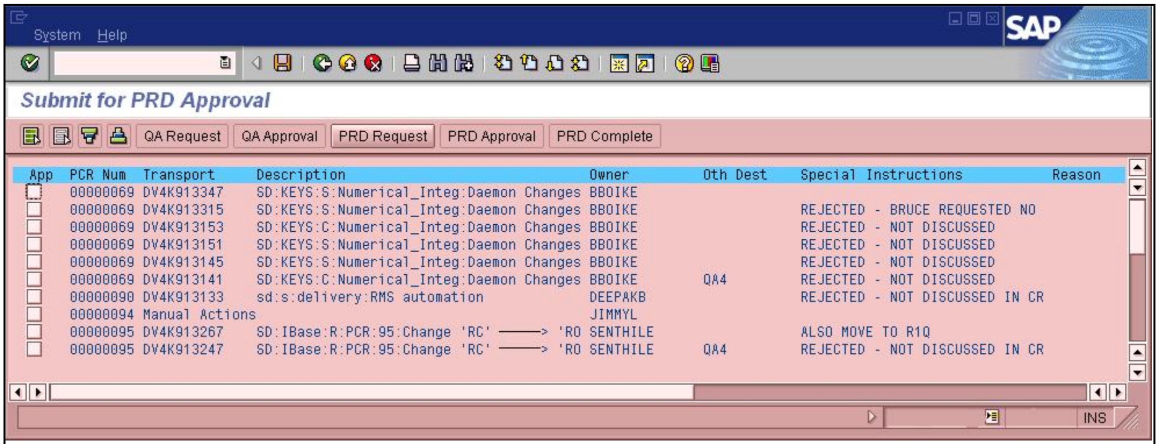
Source System: DV4

☐ Include Released TPs

User Name: ANAIR

Fig. 25. Add transport to change request.

This information is then entered into the Transport request table along with the appropriate change log in the change log table with the associated change request. The developer after doing initial testing in the development environment requests the transport to be moved to quality assurance system. The unit test document is also filled by the developer. The change request system identifies all the transports in the development systems and then lists them for the developer to request it for QA. The peer review is conducted and code review is added to change request by clicking on ‘Technical Peer Review’. The analyst who is responsible to release the transport can release the transport from the change request system. This is made possible by a function module that is remote enabled hence allowing it to be executed on any development system. This performs the release function and an automatic job that runs in the QA environment, moves the transport into the system. Figure 26 and Figure 27 shows couple of stages of CRB and each stage is explained below. It depicts the change request in different stages, from development system to production.

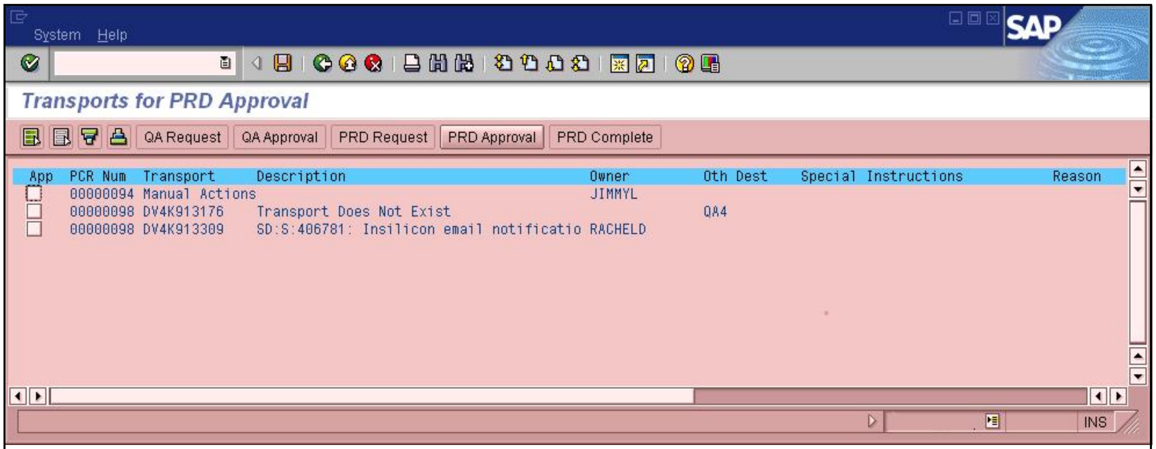


Submit for PRD Approval

| App | PCR Num | Transport | Description | Owner | Oth Dest | Special Instructions | Reason |
|---|---|---|---|---|---|---|---|
| □ | 00000069 | DV4K913347 | SD:KEYS:S:Numerical_Integ:Daemon Changes | BBOIKE | | | |
| □ | 00000069 | DV4K913315 | SD:KEYS:S:Numerical_Integ:Daemon Changes | BBOIKE | | REJECTED - BRUCE REQUESTED NO | |
| □ | 00000069 | DV4K913153 | SD:KEYS:C:Numerical_Integ:Daemon Changes | BBOIKE | | REJECTED - NOT DISCUSSED | |
| □ | 00000069 | DV4K913151 | SD:KEYS:S:Numerical_Integ:Daemon Changes | BBOIKE | | REJECTED - NOT DISCUSSED | |
| □ | 00000069 | DV4K913145 | SD:KEYS:S:Numerical_Integ:Daemon Changes | BBOIKE | | REJECTED - NOT DISCUSSED | |
| □ | 00000069 | DV4K913141 | SD:KEYS:C:Numerical_Integ:Daemon Changes | BBOIKE | QA4 | REJECTED - NOT DISCUSSED | |
| □ | 00000090 | DV4K913133 | sd:s:delivery:RMS automation | DEEPAKB | | REJECTED - NOT DISCUSSED IN CR | |
| □ | 00000094 | Manual Actions | | JIMMYL | | | |
| □ | 00000095 | DV4K913267 | SD:IBase:R:PCR:95:Change 'RC' ———> 'RO | SENTHILE | | ALSO MOVE TO R1Q | |
| □ | 00000095 | DV4K913247 | SD:IBase:R:PCR:95:Change 'RC' ———> 'RO | SENTHILE | QA4 | REJECTED - NOT DISCUSSED IN CR | |

| Buttons → | QA Request | QA Approval |
|---|---|---|
| What is done in this stage | Work in development - This is where all newly added transports appear. | Transports requested for QA import. The analyst can see a list of their transports by pressing this button. At this point they can select the ones to promote to QA by pressing the save button. |
| Who performs this | Transport Owners (Developers, Administrators) | Production Support Manager(s) will review these and approve/reject periodically. |

Fig. 26. Change request QA request and approval.

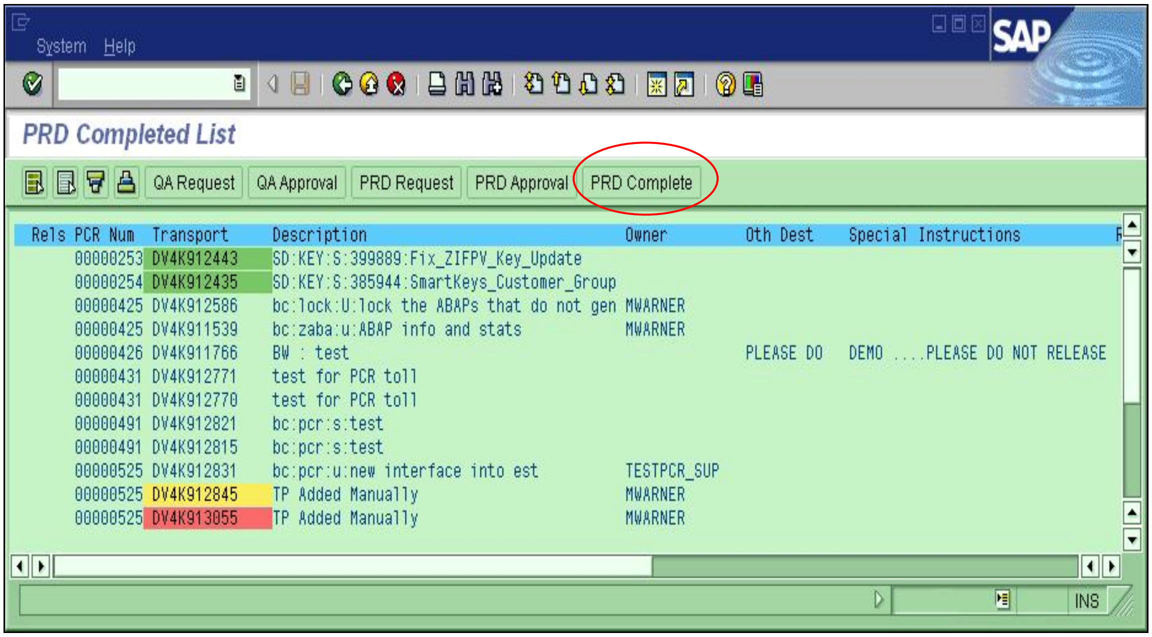


```
App  PCR Num  Transport     Description                           Owner       Oth Dest     Special Instructions          Reason
□    00000094 Manual Actions                                      JIMMYL
□    00000098 DV4K913176    Transport Does Not Exist                          QA4
□    00000098 DV4K913309    SD:S:406781: Insilicon email notificatio RACHELD
```

| Buttons → | PRD Request | PRD Approval | PRD Complete |
|---|---|---|---|
| What is done in this stage | QA testing has completed by requestor and approved by business owner. | Transports requested for PRD import. The developers can see a list of their transports by pressing this button. At this point they can select the ones to promote to PRD by pressing the save button. | List or complete transports for open change requests. |
| Who performs this | Transport Owners | Production Support Manager(s) will review these and approve/reject periodically. | All |

Fig. 27. Change requests in production approval stage.

After the transport moves to the QA system, change request system triggers an email to the requestor. The requestor then tests the changes in the QA system and approves the change request in the system. The test documents are also updated. This will trigger an email to the business approver to approve the change. The business owner approves the change and the transport is then requested for Production. The Change Review Board (CRB) then reviews all the transports weekly and then approves them to Production. The system then moves all the transport to the production system. This also changes the state of the change request according to the success of the transport in the Figure 28.



PRD Completed List

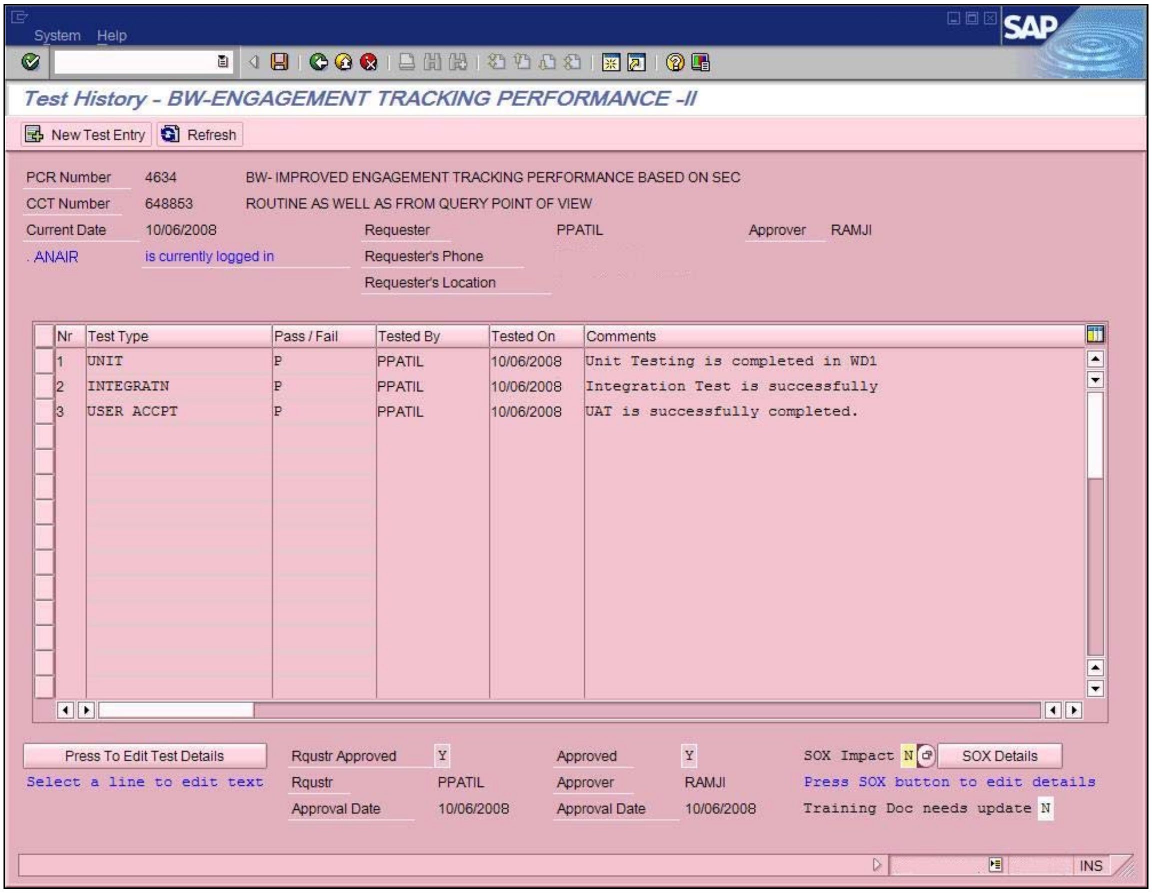
| Rels | PCR Num | Transport | Description | Owner | Oth Dest | Special Instructions |
|---|---|---|---|---|---|---|
| | 00000253 | DV4K912443 | SD:KEY:S:399889:Fix_ZIFPV_Key_Update | | | |
| | 00000254 | DV4K912435 | SD:KEY:S:385944:SmartKeys_Customer_Group | | | |
| | 00000425 | DV4K912586 | bc:lock:U:lock the ABAPs that do not gen | MWARNER | | |
| | 00000425 | DV4K911539 | bc:zaba:u:ABAP info and stats | MWARNER | | |
| | 00000426 | DV4K911766 | BW : test | | PLEASE DO | DEMO ....PLEASE DO NOT RELEASE |
| | 00000431 | DV4K912771 | test for PCR toll | | | |
| | 00000431 | DV4K912770 | test for PCR toll | | | |
| | 00000491 | DV4K912821 | bc:pcr:s:test | | | |
| | 00000491 | DV4K912815 | bc:pcr:s:test | | | |
| | 00000525 | DV4K912831 | bc:pcr:u:new interface into est | TESTPCR_SUP | | |
| | 00000525 | DV4K912845 | TP Added Manually | MWARNER | | |
| | 00000525 | DV4K913055 | TP Added Manually | MWARNER | | |

Fig. 28. Completed change request state.

- Green indicates a transport with no warning.
- No color indicates the transport has not been imported into PRD.
- Yellow indicates a transport warning.
- Red indicates a transport error.

After all the approvals and transports are complete, the “PRD Complete” button can be used to view the status of the transports. The lists that are not color coded will display all the transports belonging to change requests that are not closed. The transport number is color coded for easy problem identification.

All the change requests should have the appropriate documentation and test results. The test results are updated at each stage of the change request and entered as shown in Figure 29. Each of the test results can be reviewed by clicking the ‘Test Results’ button on the main screen of the application.



Test History - BW-ENGAGEMENT TRACKING PERFORMANCE -II

| | | | | | |
|---|---|---|---|---|---|
| PCR Number | 4634 | BW- IMPROVED ENGAGEMENT TRACKING PERFORMANCE BASED ON SEC | | | |
| CCT Number | 648853 | ROUTINE AS WELL AS FROM QUERY POINT OF VIEW | | | |
| Current Date | 10/06/2008 | Requester | PPATIL | Approver | RAMJI |
| . ANAIR | is currently logged in | Requester's Phone | | | |
| | | Requester's Location | | | |

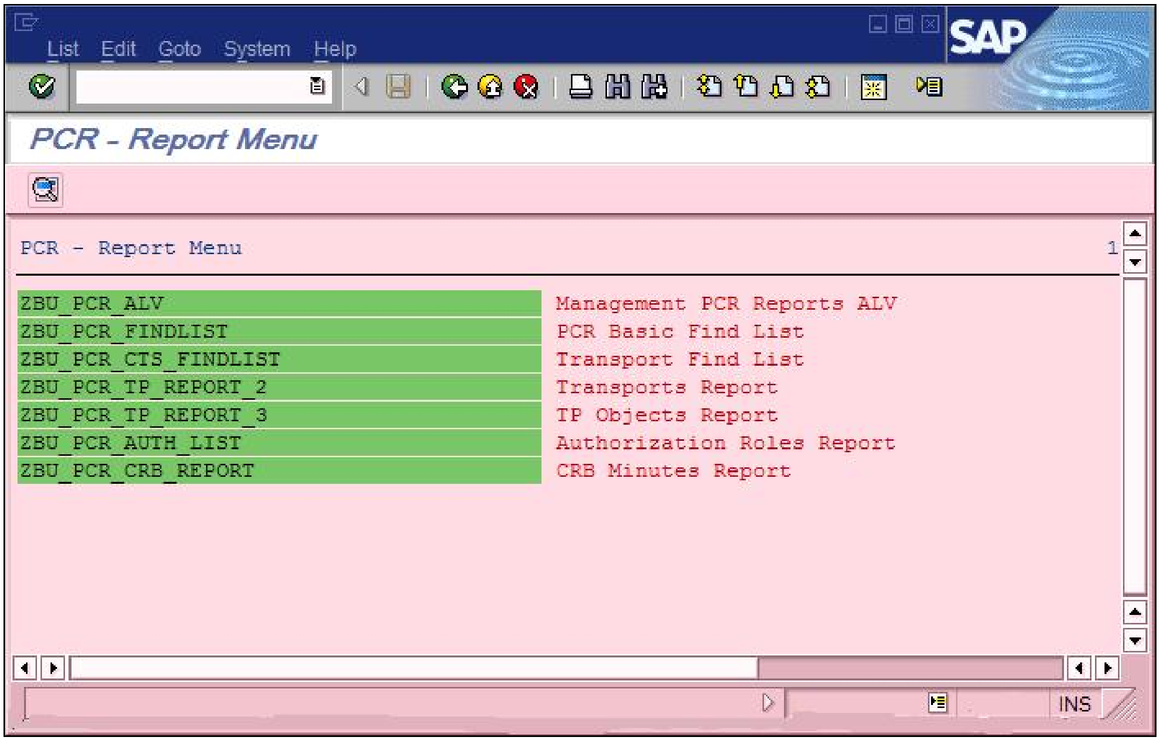
| Nr | Test Type | Pass / Fail | Tested By | Tested On | Comments |
|---|---|---|---|---|---|
| 1 | UNIT | P | PPATIL | 10/06/2008 | Unit Testing is completed in WD1 |
| 2 | INTEGRATN | P | PPATIL | 10/06/2008 | Integration Test is successfully |
| 3 | USER ACCPT | P | PPATIL | 10/06/2008 | UAT is successfully completed. |

| | | | | | |
|---|---|---|---|---|---|
| Press To Edit Test Details | Rqustr Approved | Y | Approved | Y | SOX Impact N SOX Details |
| Select a line to edit text | Rqustr | PPATIL | Approver | RAMJI | Press SOX button to edit details |
| | Approval Date | 10/06/2008 | Approval Date | 10/06/2008 | Training Doc needs update N |

Fig. 29. Test results.

The change request is now complete. All the change requests can be reviewed by the management and the auditors. For this purpose there are few reports are created and these are shown in Figure 30.

The Management PCR Reports ALV gives the report in the excel format. The data from the PCR tables and associated tables are taken and then published into an excel format that can be downloaded in many formats, like spreadsheet, word etc. Similarly, all the other reports retrieve data from the same set of change request tables and SAP standard tables to display the transports in a change request, the authorization objects



PCR - Report Menu 1

```
ZBU_PCR_ALV             Management PCR Reports ALV
ZBU_PCR_FINDLIST        PCR Basic Find List
ZBU_PCR_CTS_FINDLIST    Transport Find List
ZBU_PCR_TP_REPORT_2     Transports Report
ZBU_PCR_TP_REPORT_3     TP Objects Report
ZBU_PCR_AUTH_LIST       Authorization Roles Report
ZBU_PCR_CRB_REPORT      CRB Minutes Report
```

Fig. 30. PCR reports.

involved and security roles. All these reports suffice the requirements set by the management and auditors.

# CHAPTER VI

# CONCLUSION

The project starts by explaining transports and transport management in an SAP system. It then talks about how transports move across the landscape. The development system is the originator of the transport. The transport then moves from development system to the quality assurance system and then through production according to the transport route defined in the transport management system. The project aims at addressing the audit and management requirements that is directed at change management.

The project was designed such that developers and technical analyst can attach their transports and changes to a change request which can then be moved to production through a workflow. The transport is moved to QA system by the analyst. The changes are tested by requestor of the change request and approved by business owner. The appropriate documents are updated at each stage along with test results. The CRB then review the changes and then approves the changes to production. The project has finished development along with initial unit testing. The project is ready for UAT and will start in the first week of June.

The change request system delivers all the needs of the management and auditors. It also ensures that all the changes that go to production are reviewed properly and tested. All the changes that have gone to production in a year can be listed to view how many changes have taken place in each functional area. This also gives management an idea of the amount of changes that are getting approved for each year. It is also easy to track what enhancement, bug fixes and implementations were done by a developer. Hence all production changes are fully tested, validated, reviewed and reported.

## Future Work

Currently the software can track changes only and present the management as well as the support team the list of developments. The change management system can have a few improvements in terms of features that will give the software new dimensions.

- One of the drawbacks of SAP transport system is that it cannot revert a transport that has gone to production. The only way to fix an issue with a transport, if it occurs despite all the testing, is to send another transport to fix the issue. The change management system can use ABAP Workbench’s version management system to revert the changes.
- Before huge changes like new implementations or huge enhancements go to production, it is very difficult to understand the impact on the current production. Hence the change management system can also incorporate impact analysis as a part of change management.

# REFERENCES

# REFERENCES

[1] S.M. Bainbridge, *A Complete Guide to Sarbanes-Oxley*. Cincinnati, OH: Adams Media Corporation, 2007.

[2] L.Dale, *SAP R/3 Security for IT Auditors and Managers*. Redondo Beach, CA: Daydream Publishers, 2006.

[3] F. Gallegos, S. Senft, and C. Gonzales, *Information Technology Control and Audit*. Boca Raton, FL: CRC Press, 2004.

[4] S. McFarland Metzger and S. Roehrs, S*AP R/3 Change and Transport Management*. New York, NY: Addison-Wesley, 2003.

[5] J.A. Hernandez, *SAP R/3 Handbook*. New York, NY: McGraw-Hill Professional, 2005.

[6] U. Mende, *Software Development for SAP R/3*. New York, NY: Springer, 2000.

[7] D. Woods and J.Word, S*AP Netweaver for Dummies*. New York, NY: John Wiley & Sons, Inc. 2004.

[8] K. Greenwood, *SAMS Teach Yourself ABAP/4® in 21 Days*. Indianapolis, IN: SAMS, 1999.

[9] B. Matzke, *ABAP/4 Programming the SAP R/3 System*. New York, NY: Addison-Wesley, 2001.

[10] H. Kagermann, W. Kinney, K. Kuting, C.-P. Weber, *Internal Audit Handbook: Management with the SAP-Audit Roadmap*. New York, NY: Springer, 2008.