

HTTP based Printer Control system to print from a Mobile device

DISSERTATION REPORT

Submitted in partial fulfillment of the requirements of

M.S. Software Engineering Degree programme

by

Karamchetti Naveen

1998HZ71955

Under the supervision of

Dr. V. V. S Udaya Bhaskar

Senior Technology Consultant

Dissertation work carried out at

Wipro Technologies, Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani (Rajasthan) INDIA

May 2002

HTTP based Printer Control system to print from a Mobile device

SEWP ZG629T DISSERTATION

Submitted in partial fulfillment of the requirements of

M.S. Software Engineering Degree programme
by

Naveen Karamchetti
1998HZ71955

Under the supervision of

Dr. V. V. S Udaya Bhaskar
Senior Technology Consultant

Dissertation work carried out at
Wipro Technologies, Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA

May 2002

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled

"HTTP based Printer Control system to print from a Mobile device"

and submitted by Karamchetti Naveen ID No. 1998HZ71955

in partial fulfillment of the requirements of SEWP ZG629T Dissertation, embodies the work done by him under my supervision.

Date:

Signature of the Supervisor

Name: Dr. V.V.S Udaya Bhaskar
Designation: Senior Technology Consultant

Acknowledgements

I am very grateful to my mentor Dr. V.V.S Udaya Bhaskar for giving me valuable technology insights about how things work or rather supposed to work. His suggestions and guidance about how to make complex things look simple were very valuable, something that I would cherish for the rest of my life. His constant monitoring of the progress of work, feedback and appropriate inputs at the right time had proven invaluable towards the betterment of this work.

I would like to thank Mr. Hussain Nayak Shoukath for showing me a way, for all my problems and making sure that I never put the wrong foot forward.

I would also like to thank Mr. N Shyam Sunder for telling me where things can go wrong and also, what is wrong with things.

Further, I would also like to thank Mr Purna Chandra Rao, Mr K. Shiva Prasad and Mr D.Shiva Prasad, my ex-colleagues, who were ready to give me a helping hand whenever I needed, throughout the course of the entire project.

I sincerely thank all my friends and colleagues for their valuable contributions and inputs. I may not be able to name them all, but their assistance by way of books; pointers to resources, general discussions and clarifications have been quite useful.

I specially thank the management of Wipro Technologies for giving me this opportunity and also making available all the resources without which I could not have completed this dissertation. I am also thankful to BITS Pilani and the Training department for all their efforts in organizing this course.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

SECOND SEMESTER 2001-2002

Title of Dissertation: HTTP based Printer Control system to enable Mobile Printing.

Name of Supervisor: Dr. V.V.S Udaya Bhaskar

Name of Student: Karamchetti Naveen

ID No. of the Student: 1998HZ71955

Abstract

The popularity of mobile computing and communication devices can be traced to their ability to deliver information to users when needed. Users want ubiquitous access to information and applications from any device at hand (Mobile phone, PDA etc) plus they want to access and **print** this information on the fly. This dissertation work would try and explore one such interesting area, an untapped area in the Mobile world, a service that can prove highly useful and inseparable if realized, which is to print out a document anytime, anywhere.

The dissertation work involves the development of a printer control system, which behaves as a client. This printer control system (printer terminal) communicates with a central server, notifying its status and other Meta information. A custom communication protocol over HTTP has been defined to enable proper interaction between the printer terminal and the print server. The printer terminal is responsible for downloading the job information and printing it on the printer in addition to various other tasks. The Epson Application System (EAS), interfaces with the mobile users by providing a web based interface. The mobile users can select a document for printing using the EAS web based user interface.

This Printer Control system would ultimately be embedded into the printer thus enabling the printer to directly interact with the EAS without the help of a PC. For the purpose of the proof of concept and completeness the Printer control system has been developed and demonstrated. The other EAS services like user registration, billing etc are out of scope of this dissertation work.

Date:

Signature of the Supervisor

Signature of the Student

Important Terms, Abbreviations and Acronyms

The following are the terms, acronyms and abbreviations used in this document:

Acronym	Meaning
EPT	Epson Printer Terminal. The printer terminal that prints a document.
EAS	Epson Application Service. The Web application that services various clients.
RPR	Register Printer Request. This request type is sent by the EPT to register itself.
DPR	Delete Printer Request. This request type is sent by EPT to de-register itself.
GJF	Get Job File. This request type is used by the EPT to request the Job File.
NPS	Notify Printer Status. This request type is used by the EPT to Notify the printer status.
NJS	Notify Job Status. This request type is used by the EPT to Notify the job status.
NWJ	New Print Job. This notification is sent to the EPT by the EAS, when a new print job is initiated by the user.
JQ	Job Queue. The MSMQ component.
JQL	Job Queue Listener. The MSMQ listener.
DB Server	Database Server.
CP	Content Provider.
SP	Mobile Phone Service Provider.
DF	Document Formatter. A component that converts a given set of document types to PDF format.
PDF	Portable Document Format. A file format.
MSMQ	Microsoft Messaging Queues. A MOM technology.
SQL	Structured Query Language.

List of Figures

Id	Description	Page No
1.	iPrint System Architecture	12
2.	Value added mobile user services	16
3.	iPrint Service	17
4.	iPrint System Architecture	18
5.	Secure network of EPT's	19
6.	EAS System Architecture	21
7.	Printer registration request	29
8.	Printer deletion request	30
9.	Printer status notification request	31
10.	Job status notification request	33
11.	Get Job File request	34
12.	New Print Job	35
13.	Job Queue Controller Class Diagram	47
14.	EPrinterTerminal Class Diagram	50
15.	EPrinterTerminal Printing Class Diagram	52
16.	Deployment Diagram	60

List of Tables

Id	Description	Page No
1.	Business Requirements	14
2.	Functional Requirements	15
3.	Non-Functional Requirements	15
4.	Job Queue message type	22
5.	Printer table schema	25
6.	Paperinfo table schema	26
7.	Jobinfo table schema	26
8.	Printer Status chart	31
9.	Job Status chart	32
10.	Job Queue Message label	45
11.	Job Queue data	46
12.	EprinterTerminal Classes – I	51
13.	EprinterTerminal Classes – II	52
14.	EAS System Test cases	56
15.	Job Queue Controller Test cases	57
16.	EPT Test cases	58
17.	Development Environment	59
18.	Deployment Environment	59

TABLE OF CONTENTS

EXECUTIVE SUMMARY	10
1 INTRODUCTION	11
1.1 ABOUT WIPRO TECHNOLOGIES	11
1.2 BACKGROUND	11
2 EXISTING SYSTEM DESCRIPTION	12
2.1 CONTENT PROVIDER (CP)	13
2.2 DOCUMENT FORMATTER (DF)	13
2.3 EPSON APPLICATION SYSTEM (EAS)	13
2.4 PRINTING SERVICE PROVIDERS (PSP)	13
3 PROJECT REQUIREMENTS DESCRIPTION	14
3.1 BUSINESS REQUIREMENTS	14
3.2 STATEMENT OF REQUIREMENTS.....	15
3.2.1 <i>Functional Requirements</i>	15
3.2.2 <i>Non-functional Requirements</i>	15
3.3 SYSTEM USE CASES.....	16
3.4 NEW DEVELOPMENT OVER THE EXISTING SYSTEM	17
4 SYSTEM DESIGN	18
4.1 SYSTEM OVERVIEW	18
4.1.1 <i>Content Provider (CP)</i>	19
4.1.2 <i>Document Formatter (DF)</i>	19
4.1.3 <i>Epson Application System (EAS)</i>	19
4.1.4 <i>Epson Printer Terminal (EPT)</i>	19
4.2 EAS AND RELATED SUB-SYSTEMS	20
4.2.1 <i>Job Queue</i>	22
4.2.2 <i>Job Queue Controller</i>	22
4.2.3 <i>Web Server</i>	23
4.2.4 <i>Database Server</i>	25
4.2.5 <i>Communication between various modules</i>	27
4.2.6 <i>Design Considerations</i>	27
4.3 EAS DESIGN AND PROCESS FLOW	28
4.3.1 <i>Printer Registration (RPR)</i>	29
4.3.2 <i>Printer Deletion (DPR)</i>	30
4.3.3 <i>Printer Status Notification (NPS)</i>	31
4.3.4 <i>Job Status Notification (NJS)</i>	32
4.3.5 <i>Getting the Job File (GJF)</i>	33
4.3.6 <i>New print Job (NWJ)</i>	34
4.4 EAS COMMUNICATION PROTOCOL	36
4.4.1 <i>Request/Response/Notification Categories</i>	37
4.4.2 <i>Request formats between EPT and EAS</i>	38
4.4.3 <i>Response formats between EPT and EAS</i>	42
4.4.4 <i>Request formats between EAS and EPT</i>	44
4.4.5 <i>Response formats between EAS and EPT</i>	44
4.5 QUEUE MESSAGES	45
4.5.1 <i>Message Label</i>	45
4.5.2 <i>Message Body</i>	45
4.5.3 <i>Job Queue data</i>	46

4.6	JOB QUEUE CONTROLLER DESIGN	47
4.7	EAS START-UP PROCEDURE	48
4.8	EPT OPERATION IN A NUTSHELL.....	48
4.9	EPT DESIGN AND ARCHITECTURE	49
4.9.1	<i>EPT System Overview</i>	49
4.9.2	<i>EPT Request Process Module</i>	49
4.9.3	<i>EPT Printing Process Module</i>	52
5.	TESTING PROCESS.....	53
5.1	TESTING METHODOLOGY	53
5.1.1	<i>Software Development Process</i>	53
5.1.2	<i>Quality Process</i>	53
5.2	TEST PLAN.....	54
5.2.1	<i>Test Approach</i>	55
5.2.2	<i>Test Entry Criteria</i>	55
5.2.3	<i>Text Exit Criteria</i>	55
5.3	TEST CASES.....	55
5.3.1	<i>EAS System</i>	55
5.3.2	<i>Job Queue Controller</i>	56
5.3.3	<i>EPT</i>	57
6.	DEPLOYMENT ENVIRONMENT CONSIDERATIONS	59
6.1	DEVELOPMENT ENVIRONMENT	59
6.2	DEPLOYMENT ENVIRONMENT	59
6.3	DEPLOYMENT DIAGRAM.....	59
6.4.1	<i>Server Deployment</i>	60
6.4.2	<i>Client Deployment</i>	60
7.	CONCLUSIONS AND DEDUCTIONS	61
8.	RECOMMENDATIONS	61
8.1	SCALABILITY	61
8.2	LOAD BALANCING	61
8.3	AVAILABILITY	61
8.4	ADMINISTRATION OF SERVERS.....	62
9.	APPENDIX.....	63
9.1	EAS SQL INSTALLATION SCRIPT.....	63
9.2	A MESSAGE QUEUING ANALOGY	65
10.	REFERENCES/BIBLIOGRAPHY.....	66

Executive Summary

In recent years, wireless telecommunications have become a common subject of technical papers and area of research. There are a lot of things that are still to be explored and exploited in the areas of Mobile commerce. The new trend in technology is to provide users to have all they could possibly need in a pocket sized device. Mobility is the new buzzword in the business world. Over time, expectations have risen about what exactly it means being mobile. Mobility is the ability to access information and services from anywhere and at anytime. Some of the services that have been made possible are the services that include banking applications, online shopping, Share trading, Net browsing, Printing on LAN, PIM, E-mail, Short Messaging, Games etc. You dream of an idea that's possible in the Internet world and the same thing should be possible or can be made possible to work through a mobile. The increase in expectations of the mobile public over recent years has been driven by the rapid development of wireless technology. From mobile phones to PDA's and handheld computers, the devices being developed have become smaller and more powerful.

The popularity of mobile computing and communication devices can be traced to their ability to deliver information to users when needed. Users want ubiquitous access to information and applications from any device at hand (Mobile Phone, PDA etc) plus they want to access and **print** this information on the fly.

This dissertation work would try and explore one such interesting area, an untapped area in the Mobile World, a service that can prove highly useful and inseparable if realized, which is to print out a document anytime, anywhere.

1 Introduction

This "iPrint Service" is of great value for individuals who need any information available on the Internet as a printout in their everyday work. It is significant for those who consistently need information from web sites but cannot afford to spend more time connecting and searching and reading information from the net.

This service in a nutshell downloads the content requested by the mobile user from Content Provider using the HTTP protocol and converts the document into PDF format using Document Formatter and ultimately sends it to Printing Service Provider for printing.

1.1 About Wipro Technologies

Wipro Technologies is the global technology services division of Wipro Limited. (NYSE:WIT). Working in 20 offices around the world, we use powerful technologies to provide services for business transformation and product realization as well as complete solutions for the service provider market. Our company aims at empowering the enterprise, from getting e-businesses up and going to managing technology infrastructure, we are focused on helping businesses do their business better.

Wipro offers services in all the major areas like telecom covering internetworking, mobile computing, wireless and wired applications, e-commerce applications, CRM, and embedded systems.

The current dissertation project undertaken is part of the telecom-based projects and which also aids in gaining skills in the e-commerce areas.

1.2 Background

A printing service kiosk is a common sight in other countries like the photocopying centers in India. There are also content providers around the world supplying content to the end user through the Internet. But there should be a common channel, which integrates these two vital services, which can pull the content from the content service providers and facilitate printing with the help of the Print Service providers.

This iPrint service aids in serving this purpose. Hence this service will ease the process of taking any kind of printouts anywhere, by integrating the services of Print Service supporters and Content Providers ultimately giving the print out to the user in the preferred format.

2 Existing System Description

The iPrint service will interact with the various mobile phone users belonging to various Service Providers (SP) like I-Mode, JSky etc. The iPrint service fetches the content from the official content providers for various SP's. The protocols that can be used to fetch content can be HTTP, HTTPS or FTP. Currently, there are three content providers. They are the mail printing, horoscope printing and news printing. New Content Providers (CP) can be added when needed.

On the other side (Refer Figure 1) the iPrint Service will interact with various Printing Service Providers (PSP) such as 7DC, Lawson etc. Each PSP has various Printing Stations (PS) associated with it. The documents to be printed on any of the printing stations have to be sent through the associated PSP. Each PSP shall have an interface with the iPrint service.

The EAS also called the "Main system" interacts with the user. The EAS provides the following services to the users which include user registration, user de-registration, job status notification, printer box, job box etc.

The figure below shows the system architecture of the iPrint service

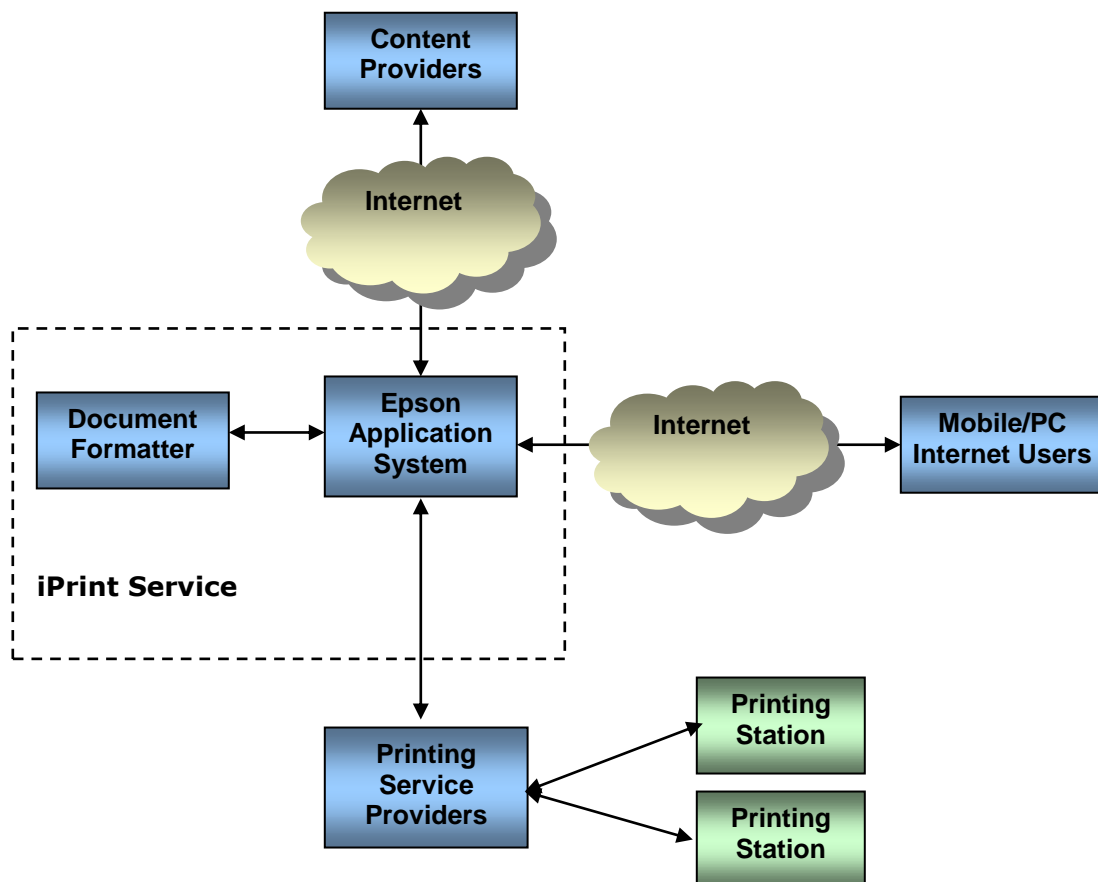


Figure 1: iPrint System Architecture

The iPrint Service consists of the following subsystems

- ❖ Content Providers (CP)
- ❖ Document Formatter (DF)
- ❖ Epson Application System (EAS)
- ❖ Print Service Providers (PSP)

2.1 Content Provider (CP)

The CP provides the content that was selected by the mobile user to the EAS. For e.g. www.yahoo.com is a CP. CP must be registered with the EAS system in order to provide content to the EAS.

2.2 Document Formatter (DF)

Document Formatter converts the user content to the common format (PDF), using the content and the formatting information supplied by user or Content Provider. Document formatter gets a message for document conversion from DF Interface. This message contains the Content to be converted and formatting information. Document Formatter converts it to the PDF format and sends notification to the DF Interface. Epson & Zeon have developed this subsystem. The PDF file is the format of the job file that will be printed on the EPT.

2.3 Epson Application System (EAS)

The EAS is primarily a web server, which serves content to either a Mobile device or a Web browser. It provides the following services to the users

- ❖ User Registration
- ❖ User De-registration
- ❖ Job Status enquiry
- ❖ Print request initiation
- ❖ Print Job cancellation
- ❖ Searching Printers

2.4 Printing Service Providers (PSP)

A Printing Service Provider is a server to which many printing stations are connected. The PSP keeps information about various printers connected to it. The print data is diverted to the printing stations based upon on the information given by the iPrint Service. The PSP informs the iPrint Service, when new printers are added or existing printers are deleted. A secure network exists between the PSP and the EAS.

3 Project Requirements Description

The system to be developed should be plug gable to the existing EAS system. A new business model is being followed for this development. It is known as the "Home Printing" model.

Highlights of the "Home Printing" model

1. Printer can be installed at public places and the shop/booth owner user can register it.
2. The application users can logically own the printers, similar to book marking web pages.
3. Every printer will have a unique id, just like a telephone number. It can be a globally unique identifier (GUID).
4. The printer will be identified based on the area where it is located. The printer id will consist of a code that identifies the area. For instance the printer id will consist of the Area Code and the Printer Id.

3.1 Business Requirements

S No.	Requirement
1.	The system should be developed as a plug gable module to the existing system.
2.	The system should support flexible deployment of software modules
3.	The system should be developed in an architectural neutral fashion so that platform dependencies can be avoided with minimal amount of changes
4.	Should require fewer amounts of changes to be done to the existing code.

Table 1. Business Requirements

3.2 Statement of Requirements

3.2.1 Functional Requirements

S No.	Requirement
1.	The EAS system shall be able to send and receive requests from a remote printer.
2.	The EAS system shall support the registration of the printer. The registration process is a one-time activity. Once the printer is registered it need not be registered again. The registration process assigns a new printer id to a printer.
3.	The EAS system shall support de-registration of the printer. If the printer capabilities are changed the printer shall be deregistered. De-registering a printer shall remove the printer from the print network. The printer shall not be visible henceforth to the mobile phone users.
4.	The EAS system shall support the notification of the printer status from the remote printer. This notification mechanism informs the status of the printer.
5.	The EAS system shall support the notification of the job status from the remote printer. This notification informs the status of the print job to the EAS.
6.	The EAS system shall have the facility to cancel the print jobs.
7.	Dummy programs shall be developed wherever it is necessary to test the system.

Table 2. Functional Requirements

3.2.2 Non-functional Requirements

S No.	Requirement
1.	The EAS system should be scalable, as the expected printers that could connect to the system could be over 1000.

Table 3. Non-Functional Requirements

3.3 System Use Cases

The primary use case identified is to provide a new value added service for mobile users in addition to the already existing services.

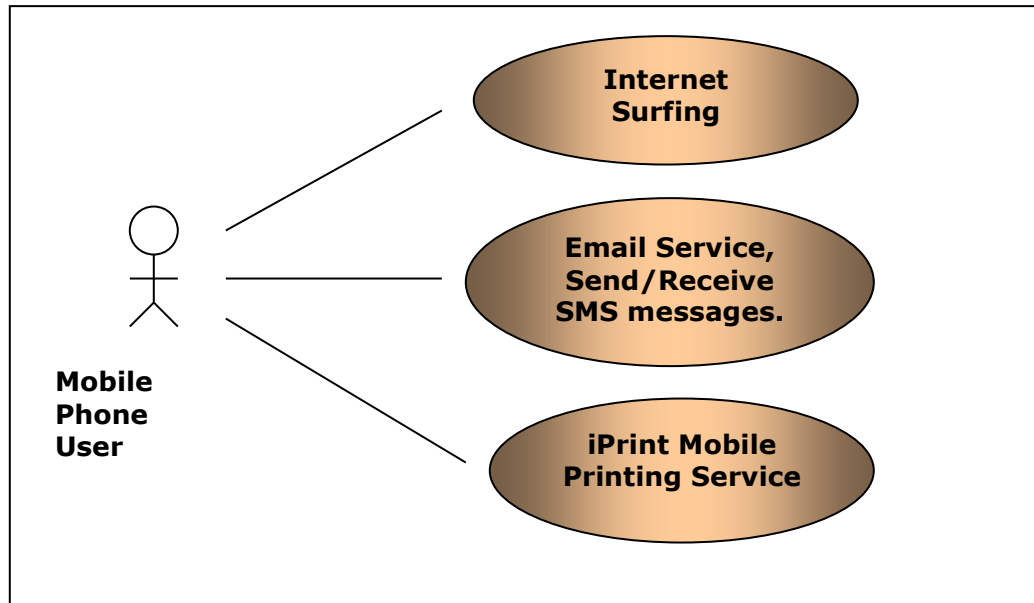


Figure 2: Value added mobile user services

The **iPrint** mobile printing service shall be a valued added service to the mobile device users. The mobile phone users shall be able to see the set of registered printers. They shall be able to select a document and then choose a printer on which the document gets printed.

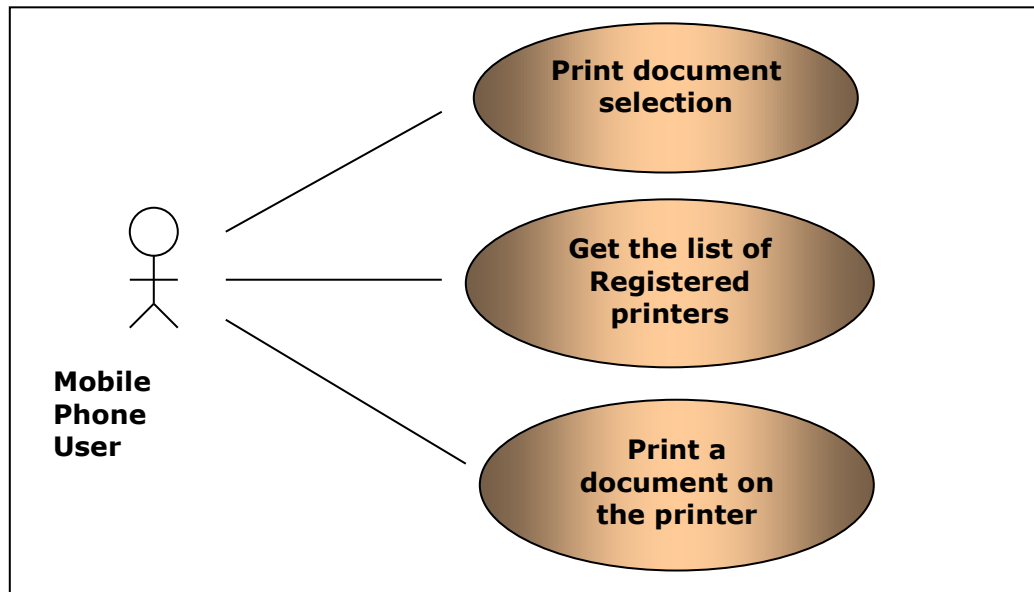


Figure 3: iPrint Service

The mobile device user shall be able to select a document and query for the list of registered printers. The user shall be able to select a printer and give his document for printing.

3.4 New development over the existing system

In the existing system the PSP was interacting with the Print stations. The PSP controlled the print stations and directed the jobs to the print stations. The print stations have no way of connecting directly with the iPrint Service.

In the new model the printer would behave as a client and would interact with the EAS system directly. This approach is known as the "Home Printing Model".

The reason for making the printer terminal to behave as a client is to have a smaller footprint for the printer and enable it to respond quickly.

One of the advantages of this model would be, to eliminate the role of PSP's and directly interact with the printer. Ultimately the printer itself will have the capability of interacting with the iPrint service without the need for a PC.

4 System Design

4.1 System Overview

The figure below shows the system architecture of the iPrint service

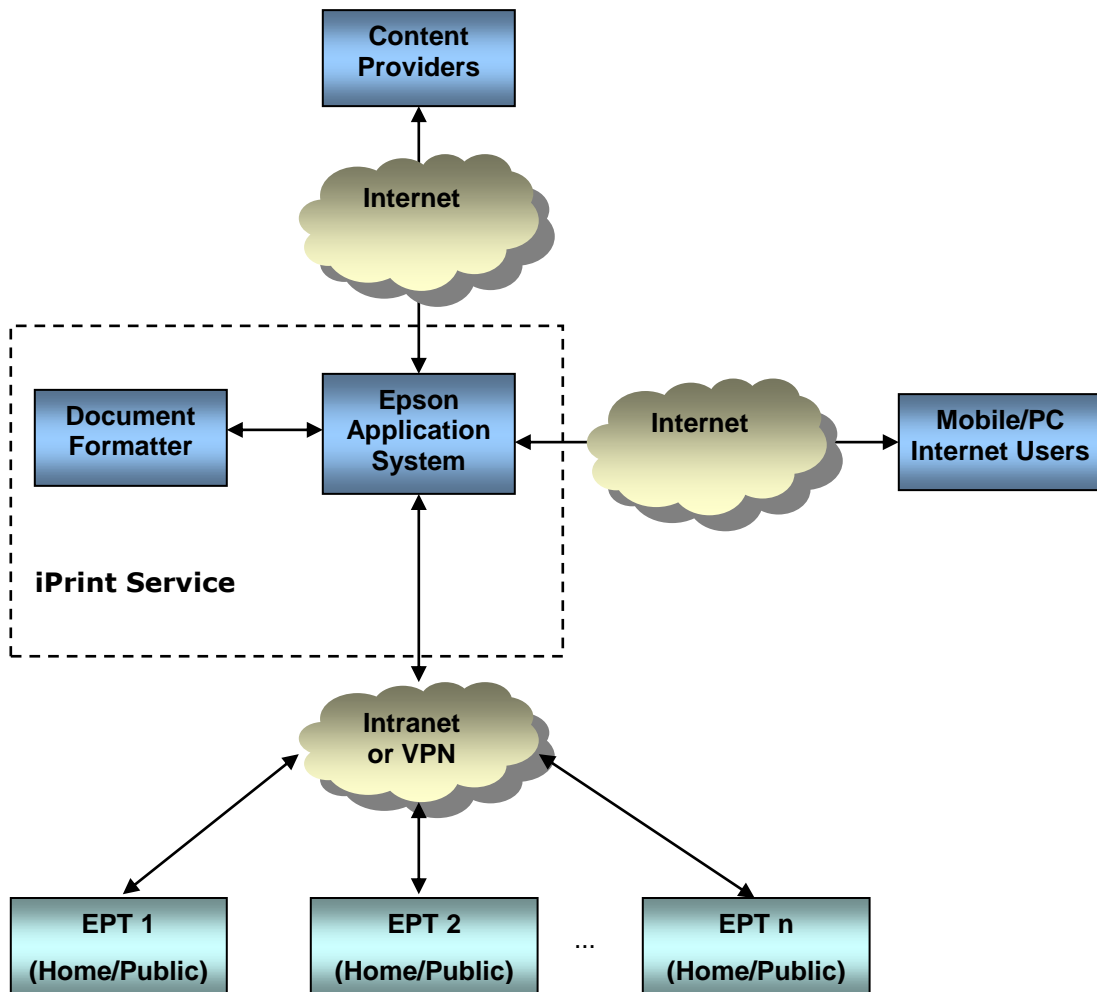


Figure 4: iPrint System Architecture

The iPrint Service consists of the following subsystems

- ❖ Content Providers (CP)
- ❖ Document Formatter (DF)
- ❖ Epson Application System (EAS)
- ❖ Epson Printer Terminal (EPT)

4.1.1 Content Provider (CP)

The CP provides the content that was selected by the mobile user to the EAS. For e.g. www.yahoo.com is a CP. CP must be registered with the EAS system in order to provide content to the EAS.

4.1.2 Document Formatter (DF)

The DF converts any web content to the Portable Document Format (PDF). The PDF file is the format of the job file that will be printed on the EPT.

4.1.3 Epson Application System (EAS)

The EAS is primarily a web server, which serves content to either a Mobile device or a Web browser. In addition to this it handles custom HTTP requests from the EPT using Active Server Pages and some other components.

4.1.4 Epson Printer Terminal (EPT)

The EPT is typically a printer, which prints out a document. This device notifies the EAS of its existence. It requests for the print job files, after getting the new print job notification from the EAS.

The EPT's operate in a secure network and communicate with the EAS directly. The need for an authentication/encryption system does not arise since the EPT's are operating in a secure network.

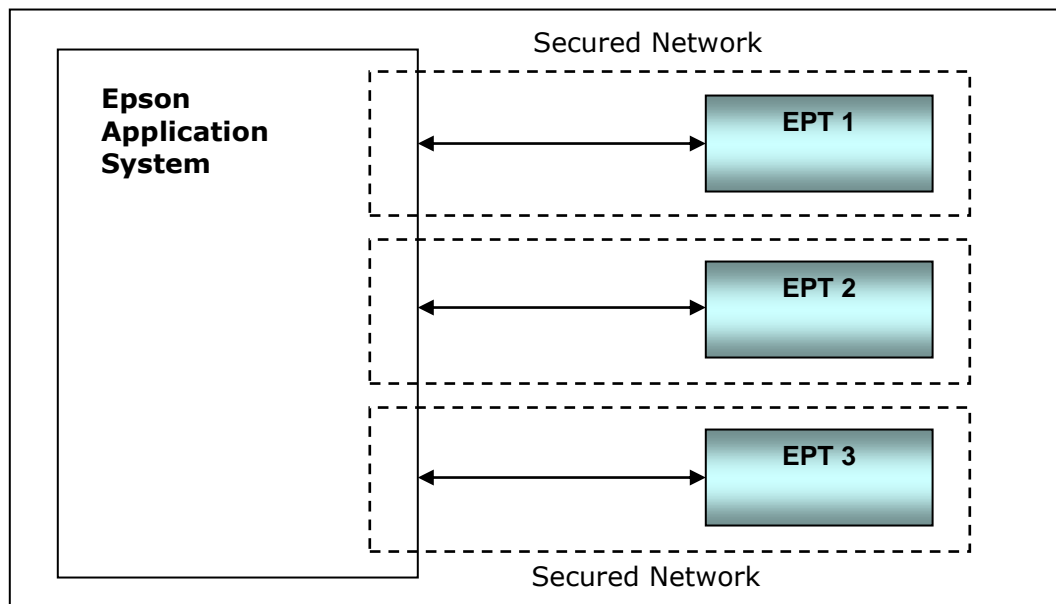


Figure 5: Secure network of EPT's

4.2 EAS and related sub-systems

Epson Application System (EAS) is a central server that interacts with the Epson Printer Terminal (EPT). EAS receives all the requests from the EPT. Based on the requests received, EAS has the knowledge of the current status of the EPT. EAS sends a response to EPT after processing the requests sent by it.

Multiple EPT's can be connected to EAS. Every EPT is identified uniquely by the EAS using a unique id, known as the <PrinterId>, which is a combination of the area code where the printer resides and a numeric value.

EPT is the actual terminal where the user can collect the printed document.

The operation of getting a document printed at EPT is explained in the following steps

- ❖ The mobile phone user selects a document to be printed.
- ❖ A selection of a printer on which, the document has to be printed is done.
- ❖ The EAS sends a notification to the EPT to get the printer status.
- ❖ If the printer status is "Ready" EAS pushes the job to the EPT and updates the status in its database. The EPT prints out the document.
- ❖ If the printer status is "Not Ready" EAS updates the status in its database. The user is notified of the status of the printer.

The operation and interaction of the EPT with the EAS is explained in the following steps

- ❖ Every EPT has to first register itself with the EAS. After the registration process, EPT shall be ready to accept print job requests from the EAS.
- ❖ As soon as the EPT comes up it sends a printer status request to the EAS specifying its current status and its IP address. The printer status information is used by the EAS to send print jobs to the EPT.
- ❖ The EPT requests the Print Job file from the EAS after it received the Print Job data
- ❖ The EPT notifies the EAS when the print job has been printed successfully on the printer.

The EAS has the following sub-modules:

- ❖ Job Queue
- ❖ Job Queue Listener
- ❖ Web Server
- ❖ Database Server

The diagram below shows the EAS system architecture:

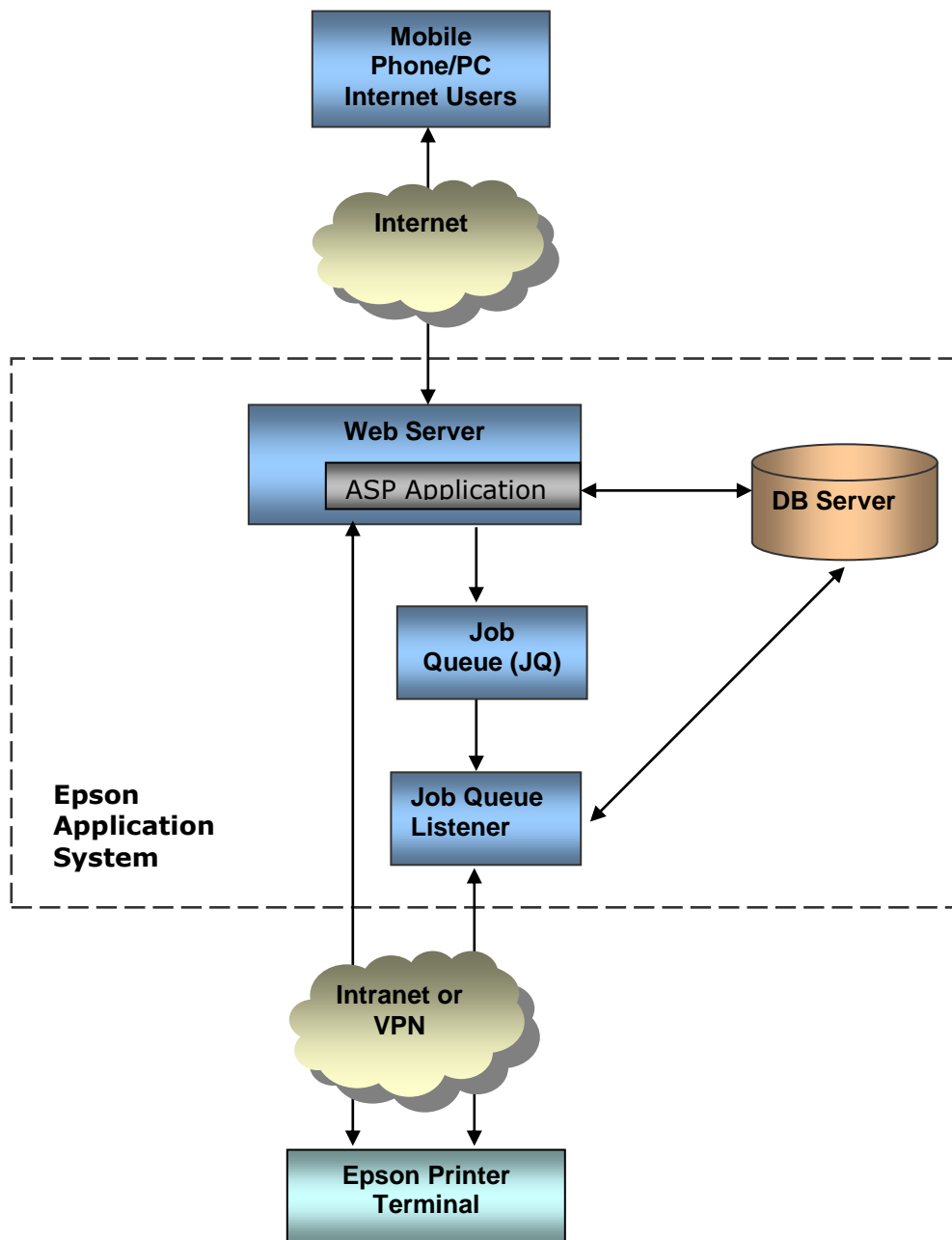


Figure 6: EAS System Architecture

4.2.1 Job Queue

The Job Queue (JQ) is used by the EAS to enable asynchronous communication. The Job Queue shall be implemented using the MSMQ technology. Message queue-based communication using Message Queuing (MSMQ) [Refer bibliography reference 3] in the Windows operating system family offers a rich feature set for applications which include, extremely fast inter-application communication, message delivery guarantees, sophisticated message and queue security, queue location transparency, rich programming language and automatic trigger support for rapid application development, and active directory integration.

EAS places various print job messages in the JQ to sent it to the EPT.

The following messages are placed by EAS in this queue:

❖ New Job (NWJ)

Message Type	Functionality
New Job (NWJ)	This message is placed by the EAS, when a new print job exists for a particular printer (EPT).

Table 4. Job Queue message type

4.2.2 Job Queue Controller

The Job Queue Controller is a listener [Refer bibliography reference 1], which looks for the messages placed by the EAS in the Job Queue (JQ). When the EAS places a message in the Job Queue (JQ), the JQ listener retrieves the message and processes it. The body of the message, which is in XML format, is retrieved and parsed. This parsed information is used to determine the EPT to which this message has to be transmitted.

When a new print job (NWJ message) is placed in the Job Queue (JQ) the Job Queue Listener picks up the message and processes it. It obtains the EPT Host IP and Port no to connect to from the message. It establishes a connection to the EPT on the Host IP and port no retrieved from the message, if the EPT is not in working condition then the message cannot be transmitted, the job status will then be changed to Failed/Retry, which can be observed by the user. If the EPT is working it sends an acknowledgement that the message has been received. When the response from the EPT is obtained, it is parsed and updated in the database.

4.2.3 Web Server

The Web server primarily serves HTML content to either a Mobile device or a Web browser. The requests from the EPT arrive at the Web server. The web server used here is Internet Information Server (IIS) [Refer bibliography reference 5]. This web server contains a set of Active Server Pages (ASP) [Refer bibliography reference 2] and additional components, which are used to receive the requests from EPT and process them. These are explained in detail in the following sub sections.

4.3.1.1 Active Server Pages (ASP) application

This ASP application [Refer bibliography reference 2] is the entry point to the EPT. All the requests from the EPT are processed by it. The EAS defines a communication protocol, which has to be followed by all EPT's. All requests to the EAS, from the EPT are in HTTP multipart format, with the body of the request in XML. The EAS validates the XML data received and processes it.

4.3.1.2 EAS Services Module

The EAS services module handles the following tasks

- ❖ Operations related to Databases.
- ❖ Operations related to MSMQ.
- ❖ Parsing and validating client XML data.
- ❖ Logging exceptions and other information.
- ❖ Generating appropriate response codes for sending responses.

4.3.1.3 Print Job Files

The Print Job files are stored in directories that are accessible to the web server. When a new print job message is sent to the JQ, the path where the files are located on the web server is also included.

The complete path where the print job files are stored is as follows:

This path contains three parts:

- a. Shared directory – This is the root directory of the shared disk. This path is stored as a configuration parameter in configuration file of the EAS system.
- b. Job Id – The Job Id of the file.
- c. File Name – The name of the print job file.

Example for a complete path for the converted pdf files:
C:\download\job_id\printme.pdf

Where, *C:\download* - is the shared directory.

\job_id – is the job id of the file to be printed.
\printme.pdf – name of the print job file.

4.3.1.4 EAS Configuration file (web.config)

The EAS configuration parameters are stored in a file named "web.config". This file contains the configuration parameters of EAS system. The configuration parameters are retrieved from this file only during startup. If the configuration information is changed at any point of time then the web server has to be restarted.

These configuration parameters can be changed, by editing the configuration file.

The various configuration parameters in this file are related to

- ❖ Job Queue name and location.
- ❖ Database URL.
- ❖ Shared directory path.
- ❖ Log file name and directory.

4.2.4 Database Server

The Microsoft SQL Server 2000 [Refer bibliography reference 7] is being used as a database server. The database server stores the following information about the printer.

- ❖ Printer id
- ❖ Printer capabilities
- ❖ Paper information
- ❖ Job information
- ❖ EPT status and other information

The following are the database tables:

Table Name	printer		
Description	Holds the information about a printer and its capabilities. The area_code and the printer_id form the primary key.		
Field Name	Data type	Length	Description
area_code - PK	String	20	The Area code where the printer is present.
printer_id - PK	String	20	The printer id of the printer.
printer_guid	String	128	The GUID of the printer.
printer_status	Char	1	The status of the printer.
printer_status_date	Date		The last known time when the printer status was updated.
manufacturer	String	30	The printer manufacturer.
model	String	30	The printer model.
type	Char	1	The category of the printer, for e.g., inkjet or laser or other
color	Char	1	The printer capability attribute.
duplex	Char	1	The printer capability attribute.
deregister	String	5	The printer de-registration flag.
EPT_host_ip	String	15	The EPT Host IP Address.
EPT_port_no	String	4	The port no. on which the EPT is listening.

Table 5: Printer table schema

Table Name	paperinfo		
Description	Holds the information about the various paper sizes supported by every printer. The area_code, printer_id and the paper_size form the primary key.		
Field Name	Data type	Length	Description
area_code – FK & PK	String	20	The area code where the printer is located.
printer_id – FK & PK	String	20	The printer id of the printer.
paper_size – PK	String	2	The standard paper size.
paper_width	String	7	The paper width.
paper_height	String	7	The paper height.

Table 6: Paperinfo table schema

Table Name	jobinfo		
Description	Holds the information about the job submitted to the printer. The area_code, printer_id and the job_id form the primary key.		
Field Name	Data type	Length	Description
area_code – FK & PK	String	20	The area code where the printer is located.
printer_id – FK & PK	String	20	The printer id of the printer.
job_id - PK	String	20	The job id of the job.
job_status	Char	1	The job status of the job.
copies	Tinyint	-	The no of copies for the job.
cost_numeric	Numeric (7,2)	-	The cost of the job.
job_date	Date	-	The date and time on which the job was started.

Table 7: Jobinfo table schema

4.2.5 Communication between various modules

- ❖ The communication between EAS and EPT is accomplished using the message oriented middleware (MOM) services [Refer bibliography reference 3]. EAS sends the requests/notifications to the job queue (JQ) as a message.
- ❖ EPT and EAS use the HTTP multi-part format for communication [Refer bibliography reference 8]. HTTP is a stateless, connection-oriented, reliable protocol. The body of the HTTP multipart request/response data format is defined in XML.
- ❖ The mobile device and the EAS communicate using the standard Internet protocols.

4.2.6 Design Considerations

- ❖ Message Queues are used for asynchronous communication in the application. Asynchronous communication is a loosely coupled approach in application integration and it is highly scalable.
- ❖ The Message Queue Listener mechanism is implemented in JQ Listener. This is implemented to avoid consuming the CPU resource until a message arrives. Since the communication between EAS and EPT is asynchronous, this mechanism is more beneficial.
- ❖ The polling mechanism is not used between EPT and EAS. Any polling mechanism consumes a lot of resources and waste CPU cycles.
- ❖ XML [Refer bibliography reference 9] is used as a communication protocol between the EAS and EPT for all the messages.
- ❖ As far as possible the no of requests coming from EPT has been reduced, this is to avoid excess and unnecessary network traffic. EPT has also some mechanism where in the print job, is pushed by the Job Queue Listener rather than EPT requesting for it.
- ❖ Authentication system is not required since the EPT's operate in a secure network. This also reduces the footprint of the EPT software.

4.3 EAS Design and Process Flow

The design and process flow is explained below according to the functionality and communication between all the modules. The request/response/notification formats are categorized into 2 types based on the modules between which the communication is taking place.

EPT to EAS

The Request/Response types from EPT to EAS

- ❖ RPR – Register Printer Request
- ❖ DPR – Delete Printer Request
- ❖ GJF – Get Job File
- ❖ NJS - Notify Job Status
- ❖ NPS –Notify Printer Status

EAS to EPT

The Request/Response types from EAS to EPT

- ❖ NWJ – New Job

4.3.1 Printer Registration (RPR)

When a new printer needs to be registered with the EAS, EPT sends the printer registration request along with printer capabilities information to the EAS. The EAS processes the request and updates the information in the database. This request is also sent by the EPT when the printer capabilities information is changed at the EPT that has to be updated in the EAS. In that case, the EPT first sends a De-register printer request (DPR) followed by a register printer request (RPR).

4.3.1.1 EPT-EAS Communication

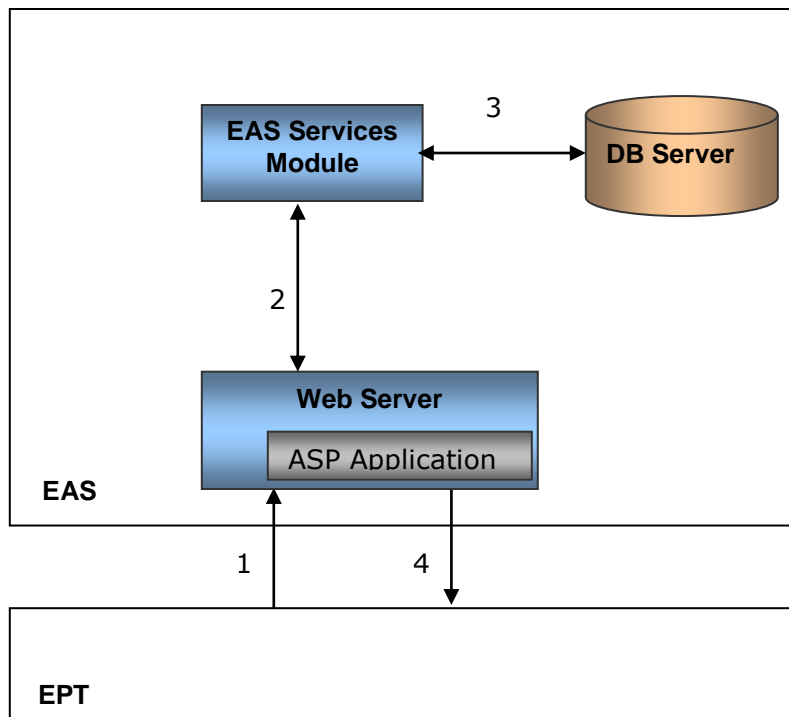


Figure 7. Printer registration request

The processing steps for printer registration request are as follows:

1. The EAS web server receives the printer registration request in the HTTP multi-part format from the EPT.
2. The XML data from HTTP multi-part body is parsed by the EAS. Taking the <AreaCode> from the request and the maximum numeric value assigned to that area code from the database, a new <PrinterId> is generated. For instance, if the area code is JBH and the maximum <PrinterId> is 350. The new <PrinterId> would be JBH_351 (350 + 1).
3. The <PrinterId> along with the printer capabilities and paper information is inserted into the database.
4. The EAS web server forms the RPR response, which also contains the return code <RC>. The printer registration response is then sent to the EPT.

4.3.2 Printer Deletion (DPR)

When a registered printer has to be unregistered, then EPT sends the printer de-registration request to EAS.

4.3.2.1 EPT-EAS Communication

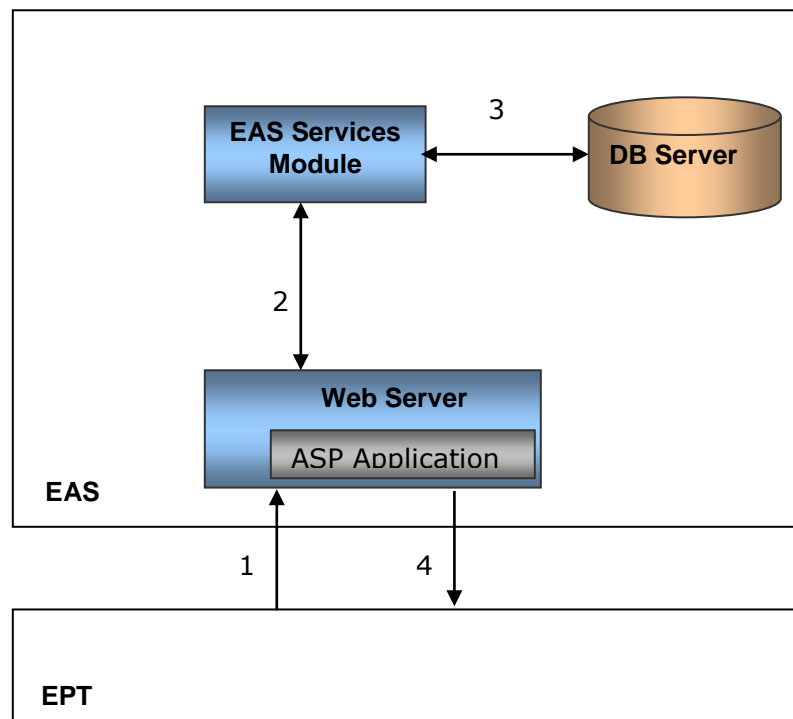


Figure 8. Printer deletion request

The processing steps for printer deletion request are as follows:

1. The EAS web server receives the printer de-registration request in the HTTP multi-part format from the EPT.
2. The XML data from HTTP multi-part body is parsed by the EAS.
3. The <PrinterId> along with the printer capabilities and paper information is retained in the database, but a flag is set in the database table which indicates that the printer is de-registered. This is done for bookkeeping information.
4. The EAS web server forms the DPR response, which also contains the return code <RC>. The printer de-registration response is then sent to the EPT.

4.3.3 Printer Status Notification (NPS)

4.3.3.1 Printer status chart

Following table describes all the printer status values sent by EPT.

Status Code	Status	Printer State
0	Not Ready	Not Ready
1	Ready	Ready

Table 8. Printer Status chart

4.3.3.2 EPT-EAS Communication

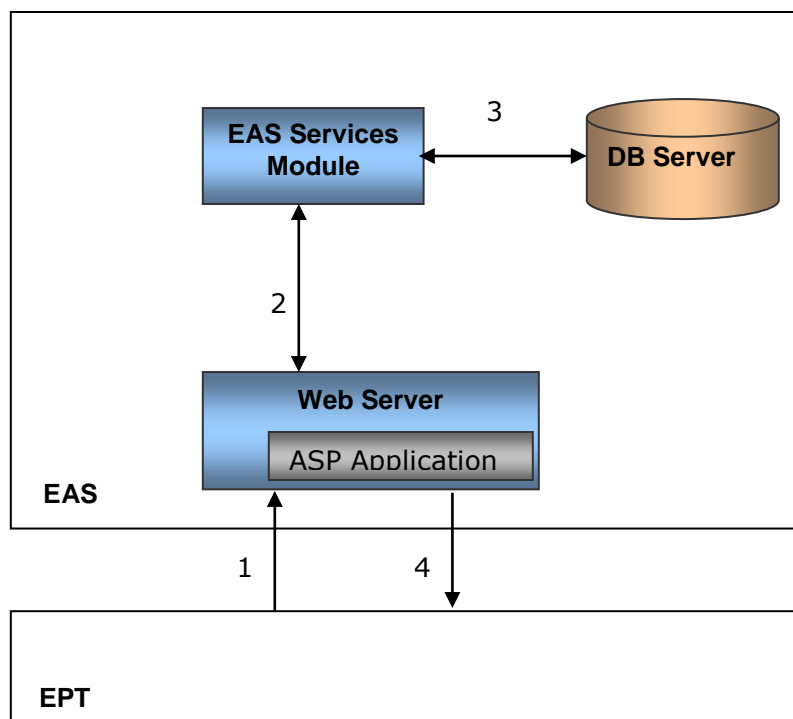


Figure 9. Printer status notification request

The processing steps for printer status notification request are as follows:

1. The EAS web server receives the Notify Printer Status (NPS) request in the HTTP multi-part format from the EPT.
2. The XML data from HTTP multi-part body is parsed by the EAS.
3. The database is updated with the new printer status information.
4. The EAS web server forms the NPS response, which also contains the return code <RC>. The notify printer status response is then sent to the EPT.

4.3.4 Job Status Notification (NJS)

4.3.4.1 Job status chart

The various status codes represent the various job states that a print job has at any point of time. Some of the codes are sent by EPT and some are set by the EAS system it self.

The following table comprises of job status and corresponding job state.

Status Code	Status	Job State
1	Success	The file specified as part of new job has completed printing.
2	Processing	Indicates that a NWJ message has been placed in the Job Queue (JQ).
3	Printing	Indicates that the EPT has just started to print the job.
4	Retry/Failed	Indicates that the EPT has failed to print the job.
5	Cancelled	Indicates that the Job has been cancelled.

Table 9. Job status chart

4.3.4.2 EPT-EAS Communication

The processing steps for job status notification request are as follows:

1. The EAS web server receives the Notify Job Status (NJS) request in the HTTP multi-part format from the EPT.
2. The XML data from HTTP multi-part body is parsed by the EAS.
3. The database is updated with the new job status information.
4. The EAS web server forms the NJS response, which also contains the return code <RC>. The notify job status response is then sent to the EPT.

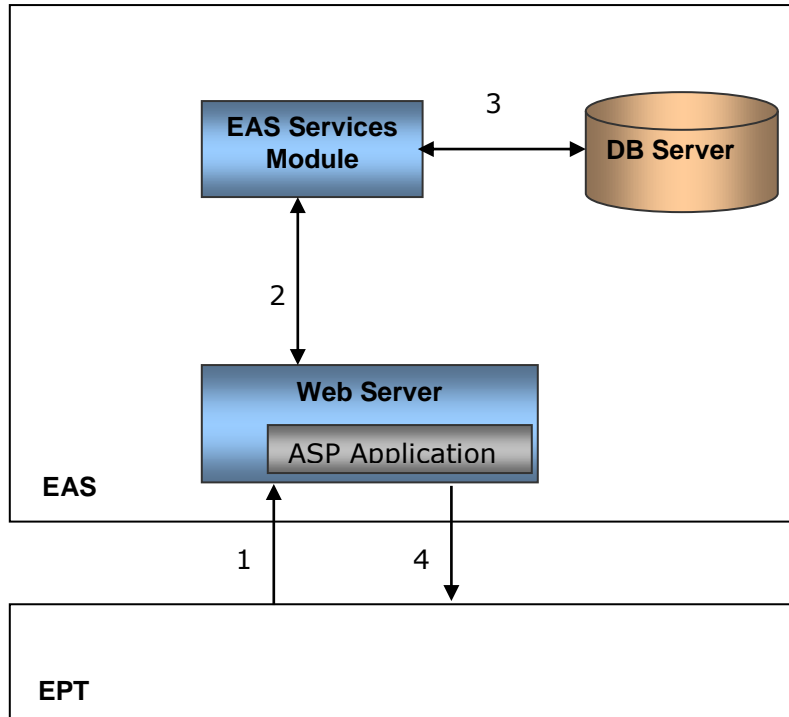


Figure 10. Job status notification request

4.3.5 Getting the Job File (GJF)

EPT sends this request after getting the job information, from the NWJ notification. With this job information, EPT requests for the file content of the print job from the EAS.

4.3.5.1 EPT-EAS Communication

The processing steps for getting the job file request is as follows:

1. The EAS web server receives the new job request in HTTP multi-part format from EPT.
2. The XML data from HTTP multi-part body is parsed by the EAS.
3. A service class returns the path of the job file from the shared disk.
4. The file content, read from the shared disk, is sent to the EPT.

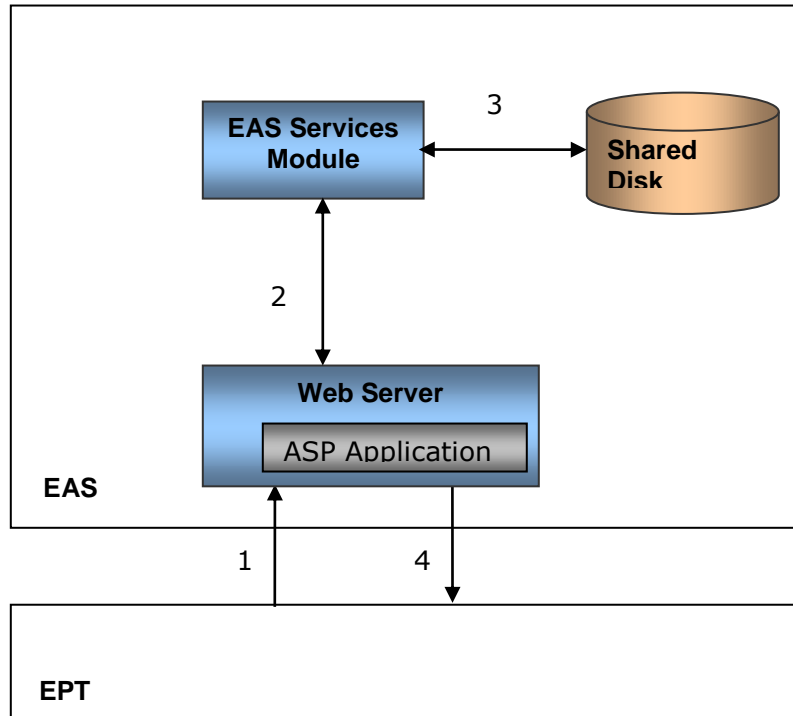


Figure 11. Get Job File request

4.3.6 New print Job (NWJ)

EAS places this message in the Job Queue to send it, to the EPT. The Job Queue Listener picks this up and transmits it to the EPT. EPT sends an acknowledgement to this print job request message.

4.3.6.1 EAS-EPT Communication

The processing steps for sending a new print job is as follows:

1. The EAS web server places a new print job request in the Job Queue (JQ).
2. The Job Queue listener picks up the message and parses the data.
3. It connects to the EPT using the information from the message and transmits the NWJ request to the EPT.
4. EPT confirms that it has received the request by sending an acknowledgement.

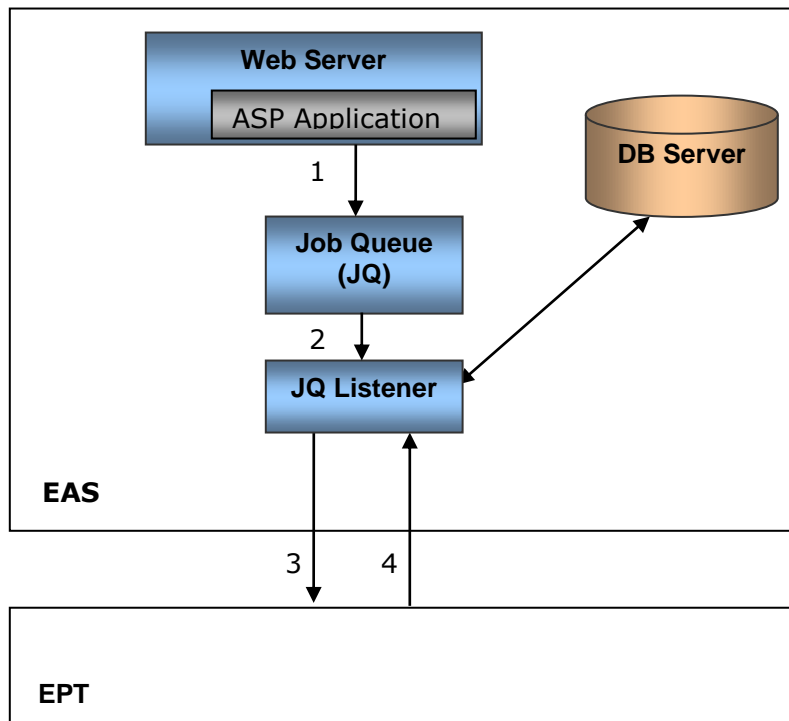


Figure 12. New Print Job

4.4 EAS Communication Protocol

This section describes the communication protocol between the Epson Application System (EAS) and the Epson Printer Terminal (EPT). The EPT developers can use this section to ensure that they are in compliance with the EAS protocol.

The following describe the conditions and constraints of the formats:

1. The formats in the request and response body are case sensitive.
2. The <PrinterId> and <Request> are mandatory in all requests. In case of register printer request (RPR) the <PrinterId> does not exist.
3. The Content-length header in the request format is mandatory. The Content-length header should contain the length of the body of the request, as per HTTP protocol [Refer bibliography reference 8].
4. In all the requests, empty lines and white spaces are ignored while parsing the body of the request.
5. The body part of the HTTP request contains valid XML data.
6. EAS will validate for identifier names, values and structure of the data.
7. The request data should be encapsulated between the starting and ending boundaries.
8. The starting boundary will have two hyphens (--) in the beginning and the ending boundary will have two hyphens (--) both at the beginning and at the end.
9. The response headers can be different depending on the web server used during deployment. The following headers will be present in any HTTP response

- ❖ Connection:
- ❖ Content-Type:
- ❖ Content-Length:
- ❖ Date:
- ❖ Server:

4.4.1 Request/Response/Notification Categories

The request/response/notification formats are categorized into 2 types based on the modules between which the communication is taking place.

EPT to EAS

The Request/Response types from EPT to EAS

- ❖ RPR – Register Printer Request
- ❖ DPR – Delete Printer Request
- ❖ GJF – Get Job File
- ❖ NJS - Notify Job Status
- ❖ NPS –Notify Printer Status

EAS to EPT

The Request/Response types from EAS to EPT

- ❖ NWJ – New Job

4.4.2 Request formats between EPT and EAS

RPR Request
POST /eas/eas.aspx HTTP/1.1 Host: easServer.com Content-type: multipart/form-data; charset=ISO-8859-1; boundary=12345 Content-length: nn Connection: close --12345 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?> <PrinterTerminal> <Request>RPR</Request> <PrinterId></PrinterId> <PrinterCap manufacturer="" model="" type="" color="" duplex=""> <PaperInfo size="" width="" height=""/> </PrinterCap> <EPTHostIp>127.0.0.1</EPTHostIp> <EPTPortNo>8080</EPTPortNo> <AreaCode>JBH</AreaCode> </PrinterTerminal> --12345--

RPR request validations			
Attribute	Expected Value	Mandatory/ Optional (M/O)	Remarks
manufacturer	String	O	The manufacturer name of the printer. Allowed maximum length is 30 characters.
model	String	M	The model name of the printer. Allowed maximum length is 30 characters.
type	String (0/1/2)	M	0 - Others 1 - Inkjet 2 - Laser
color	String (0/1)	M	0 - false 1 - true
duplex	String (0/1)	M	0 - false 1 - true
size	String	M	Predefined paper sizes. (E.g. A4)
width	Integer as String	O	Ex: 120
height	Integer as String	O	Ex: 320

DPR Request

```
POST /eas/eas.aspx HTTP/1.1
Host: easServer.com
Content-type: multipart/form-data; charset=ISO-8859-1; boundary=12345
Content-length: nn
Connection: close

--12345
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<PrinterTerminal>
<Request>DPR</Request>
<PrinterId>JBH_101</PrinterId>
</PrinterTerminal>
--12345--
```

GJF Request

```
POST /eas/eas.aspx HTTP/1.1
Host: easServer.com
Content-type: multipart/form-data; charset=ISO-8859-1; boundary=12345
Content-length: nn
Connection: close

--12345
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<PrinterTerminal>
<Request>GJF</Request>
<PrinterId>JBH_101</PrinterId>
<JobInfo jobId="" jobStatus="" />
<FileInfo name="" />
</PrinterTerminal>
--12345--
```

GJF request validations

Attribute	Expected Value	Mandatory/ Optional (M/O)	Remarks
jobId	Long as String	M	The job id of the print job.
jobStatus	Integer as String	O	
name	String	M	The print job file name.

NJS Request

POST /eas/eas.aspx HTTP/1.1

Host: easServer.com

Content-type: multipart/form-data; charset=ISO-8859-1; boundary=12345

Content-length: nn

Connection: close

--12345

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>

<PrinterTerminal>

<Request>NJS</Request>

<PrinterId>JBH_101</PrinterId>

<JobInfo jobId="" jobStatus="" />

</PrinterTerminal>

--12345—

NJS request validations

Attribute	Expected Value	Mandatory/ Optional (M/O)	Remarks
jobId	Long as String	M	
jobStatus	Integer as String (1/2/3/4/5)	M	1 - Success 2 - Processing 3 - Printing 4 - Failed/Retry 5 - Cancelled

NPS Request

POST /eas/eas.aspx HTTP/1.1

Host: easServer.com

Content-type: multipart/form-data; charset=ISO-8859-1; boundary=12345

Content-length: nn

Connection: close

--12345

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>

<PrinterTerminal>

<Request>NPS</Request>

<PrinterId>JBH_101</PrinterId>

<PrinterInfo printerStatus="" />

<EPTHostIp>127.0.0.1<EPTHostIp>

<EPTPortNo>8080< EPTPortNo>

</PrinterTerminal>

--12345—

NPS request validations

Attribute	Expected Value	Mandatory/ Optional (M/O)	Remarks
printerStatus	Integer as string (0/1)	M	0 – Not Ready 1 – Ready

4.4.3 Response formats between EPT and EAS

Response Format for RPR, DPR, NJS, NPS
--

<pre>HTTP/1.1 200 OK Server: xxx Date: Wed, 26 Sep 2001 04:59:07 GMT Content-type: multipart/form-data; charset=ISO-8859-1 Content-length: nn Connection: close <?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?> <PrintServer> <RC>200</RC> <Request>RPR/DPR/NJS/NPS</Request> <PrinterId>JBH_101</PrinterId> </PrintServer></pre>
--

GJF Response

<pre>HTTP/1.1 200 OK Server: xxx Date: Sat, 22 Sep 2001 13:43:48 GMT Content-type: application/octet-stream Content-length: nn Connection: close File content starts here... . . .</pre>

Note:

The body part of the GJF response contains the file data.

Error Response Format for all requests

HTTP/1.1 200 OK

Server: xxx

Date: Wed, 26 Sep 2001 04:59:07 GMT

Content-type: multipart/form-data; charset=ISO-8859-1

Content-length: nn

Connection: close

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>

<PrintServer>

<RC>RC No. [Return codes are in the table below]</RC>

<Request>RPR/DPR/NPS/NJS/GJF</Request>

<PrinterId>JBH_001</PrinterId>

</PrintServer>

Return Code (RC)	Detailed Informational Message
200	OK
300	Invalid XML format
301	Invalid request type
302	Invalid printer id
303	Invalid printer capability
304	Invalid paper information
305	Invalid job information
306	Invalid file information
307	Invalid printer status information
401	Unable to Register Printer
402	Unable to Delete Printer
404	Print Job Cancelled
500	EAS Internal Server Error
911	Connection to EAS failed

NOTE:

The return codes list is subject to change. New error codes will be added as and when the need arises.

4.4.4 Request formats between EAS and EPT

NWJ Request

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<PrintServer>
<Request>NWJ</Request>
<PrinterId>JBH_101</PrinterId>
<JobInfo jobId="" cost="" copies="" />
<FileInfo name="" size="" />
<EPTHostIp>127.0.0.1<EPTHostIp>
<EPTPortNo>8080< EPTPortNo>
</PrintServer>
```

4.4.5 Response formats between EAS and EPT

NWJ Response

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<PrinterTerminal>
<RC>200</RC>
<Request>NWJ</Request>
<PrinterId>JBH_101</PrinterId>
</PrinterTerminal>
```

4.5 Queue Messages

This section discusses about the message formats placed in the Queue. The Microsoft Messaging Queues (MSMQ) is used for the purpose of development.

The Message objects placed in the Job Queue (JQ) consists of two parts:

- ❖ **"Message Label"** section - holds the type of the message.
- ❖ **"Message Body"** section – holds the contents of the message.

The Message objects placed in the Job Queue (JQ) is of the type:

- ❖ **Request** – This type of message is sent to the EPT.

4.5.1 Message Label

The message labels for the Job Queue (JQ) are described below:

Job Queue Message Label

Message Label	Meaning
NWJ	New Print Job

Table 10. Job Queue Message Label

4.5.2 Message Body

The contents of the message body vary depending on the type of the message. The message body contains the Request/Datagram message in XML. This is similar to the XML data that is sent by the EPT.

If the message is of type Request, then the body part of the message contains the parameters related to the request.

The communication between EPT and EAS is in the form of XML messages with the formats similar to the EAS and EPT communication protocol as mentioned above.

4.5.3 Job Queue data

This table contains the messages that are placed in the Job Queue (JQ) by the EAS. This queue contains the request messages to EPT.

Message Name	Message Label	Message Body
New Print Job	NWJ	<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?> <PrintServer> <Request>NWJ</Request> <PrinterId>JBH_101</PrinterId> <JobInfo jobId="123456789" cost="122" copies="1" /> <FileInfo name="xyz.pdf" size="12345"/> <EPTHostIp>127.0.0.1<EPTHostIp> <EPTPortNo>8080< EPTPortNo> </PrintServer>

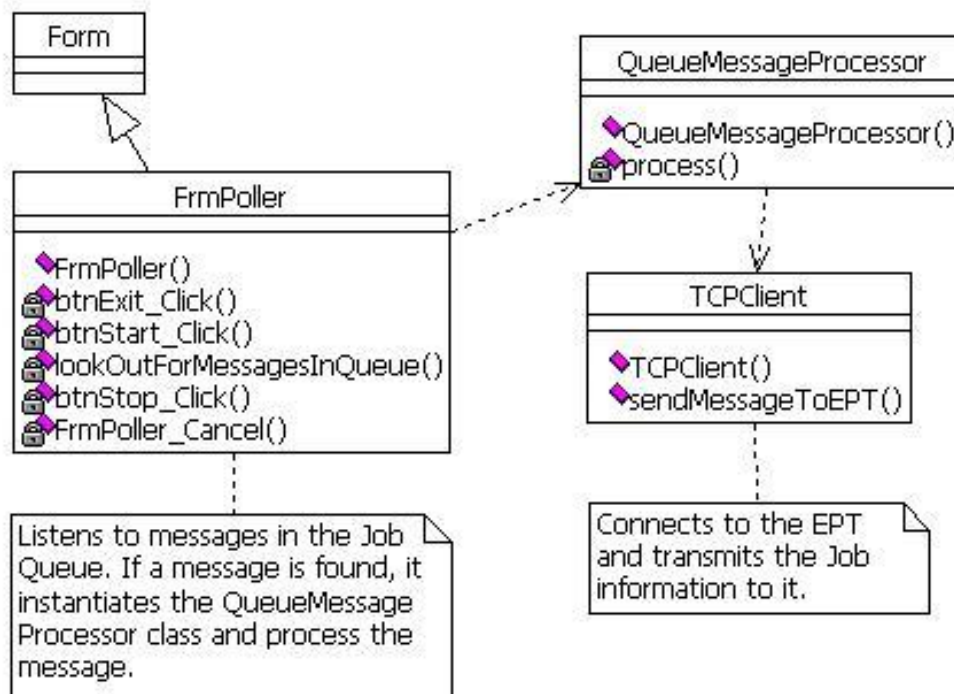
Table 11. Job Queue data

The above XML messages are generated by the EAS, and placed into the Job Queue (JQ).

NWJ Message:

This message is placed is placed in the JQ when there is a new print job for the printer.

4.6 Job Queue Controller Design



Job Queue Controller Class Diagram

Figure 13: Job Queue Controller Class Diagram

When the Job Queue Controller (JQC) starts it reads from the configuration file the job queue name and path to connect to. After the initialization process the Job Queue controller listens for messages placed in the Job Queue.

The above class diagram depicts the various classes that are used by the Job Queue controller. When the JQC starts an instance of the "FrmPoller" is created. This is the controller class for the entire application. It spawns a background thread and executes the method "lookOutForMessagesInQueue()".

The QueueMessageProcessor class is responsible for parsing the message body and retrieving the necessary information to process and transmit the message.

When a message arrives in the Job Queue, the background thread picks up the message and parses the XML data in the message body. It retrieves the EPT IP and Port no and tries to establish the connection to the EPT using the "TCPClient" class. If the connection to the EPT is not established and the message is not transmitted then the database is updated with the job status as failed/retry.

4.7 EAS Start-up procedure

- ❖ EAS essentially consists of the following components
 - Web server (Internet Information Server 5.0)
 - Database server (SQL Server 2000)
 - Messaging Queue system (Microsoft Messaging Queue)
 - Job Queue Controller (MSMQ Listener)
- ❖ For the EAS to function correctly all the above components should be started.
- ❖ When the EAS is started the following operations usually take place
 - The configuration file "web.config" is read which is under the application root directory.
 - The configuration file consists of the messaging queue location, database location, log file name and path along with other parameters.
 - The Job Queue controller is started and it starts listening for messages.
 - The EAS is now ready to accept requests from the EPT.

4.8 EPT Operation in a nutshell

- ❖ When the EPT starts up, it checks whether it is registered or not.
 - If it is registered it sends a Notify Printer Status (NPS) request to the EAS to inform the EAS regarding its presence.
 - If not, it prompts the user to register the printer. It sends a Register Printer Request (RPR) to the EAS.
- ❖ When the EAS has a job to print it sends a New Print Job (NWJ) request to the EPT, the EPT responds to the NWJ with an acknowledgment.
- ❖ The EPT then requests for the print job from the EAS. After it has received the print job it prints out the document. After the EPT prints the job it notifies the EAS.

4.9 EPT Design and architecture

This section explains the design, system overview and control of the EPT.

4.9.1 EPT System Overview

In a nutshell, EPT is an application that prints a document onto the printer. It is an application that is a layer above the printer. The main functionality of the EPT is to print a document and also notify the EAS.

The EPT application consists of two main modules, one that notifies the various statuses to the EAS, and another one that receives the print job requests from the EAS. The EPT application is both a client and server, although the server layer is thin.

4.9.2 EPT Request Process Module

EPT sends requests to the EAS for the following reasons

- ❖ To register a printer
- ❖ To deregister a printer
- ❖ To notify the job status
- ❖ To notify the printer status
- ❖ To get the print job file

The "FrmEPrinterTerminal" is the controller class for the entire application. When the application is first started, an instance of the controller class is created.

The controller class reads these configuration files and checks to see whether the printer terminal is registered or not. If the printer terminal is registered it sends a Notify Printer Status (NPS) request to the application. This informs the EAS that the printer is ready to accept print jobs.

If the printer terminal is not registered it prompts the user to register the printer. Since this printer is not registered with the EAS, it cannot receive print jobs.

The Controller class is a multithreaded application; it spawns a thread for registering a printer, it frees the UI to perform other operations. When the printer is registered the "FrmRegisterPrinter" is instantiated, this displays a user interface for the user to enter the printer registration details. When the user completes the registration form the XML data is sent to the EAS, using the "ConnectionManager".

The "ConnectionManager" establishes a TCP/IP connection to the EAS and transmits the data to the EAS. The reply got from the EAS is sent to the "XmlResponseParser" which parses the XML data and gets the Return code sent by the EAS.

The class diagram of the entire process is shown below...

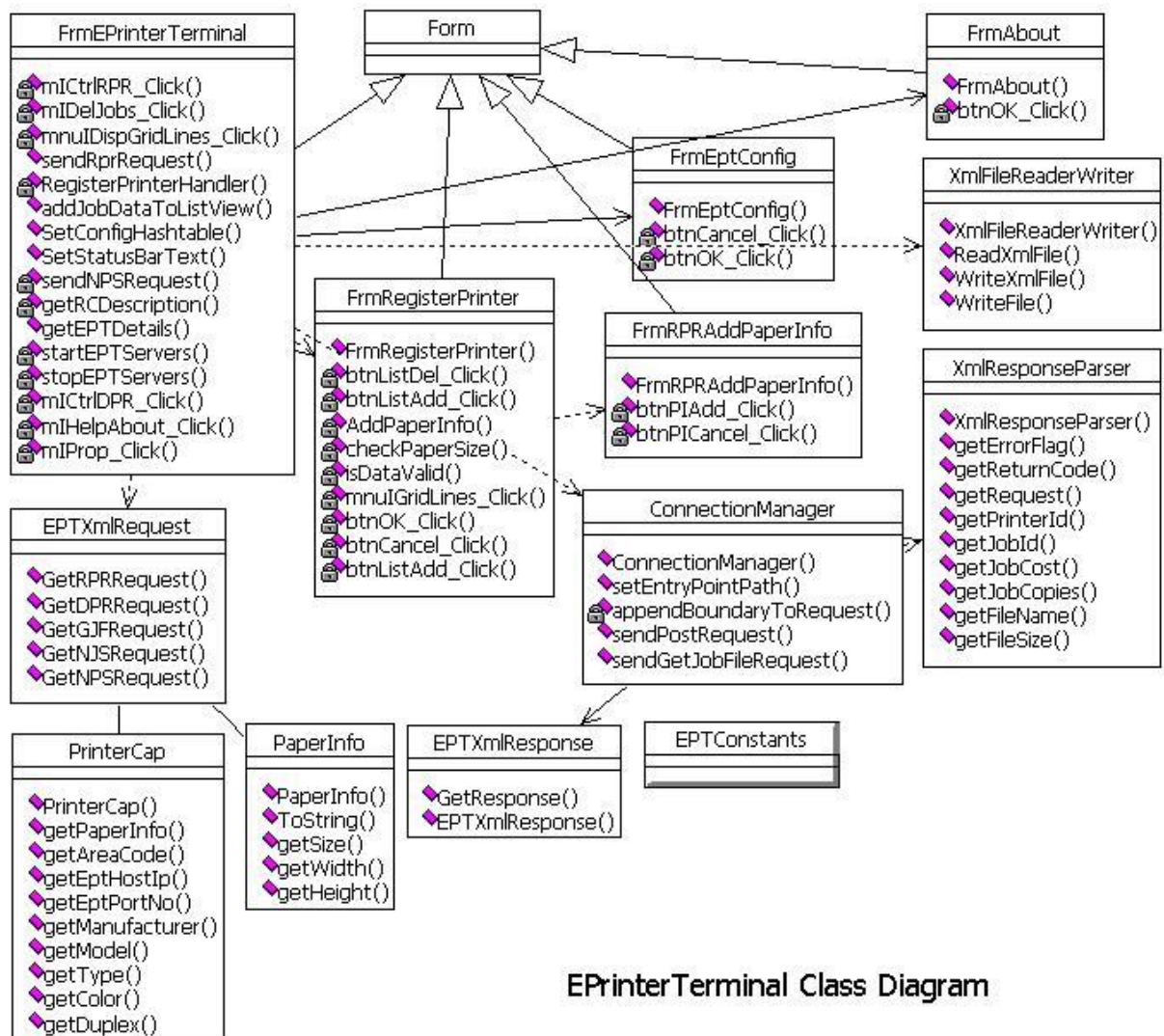


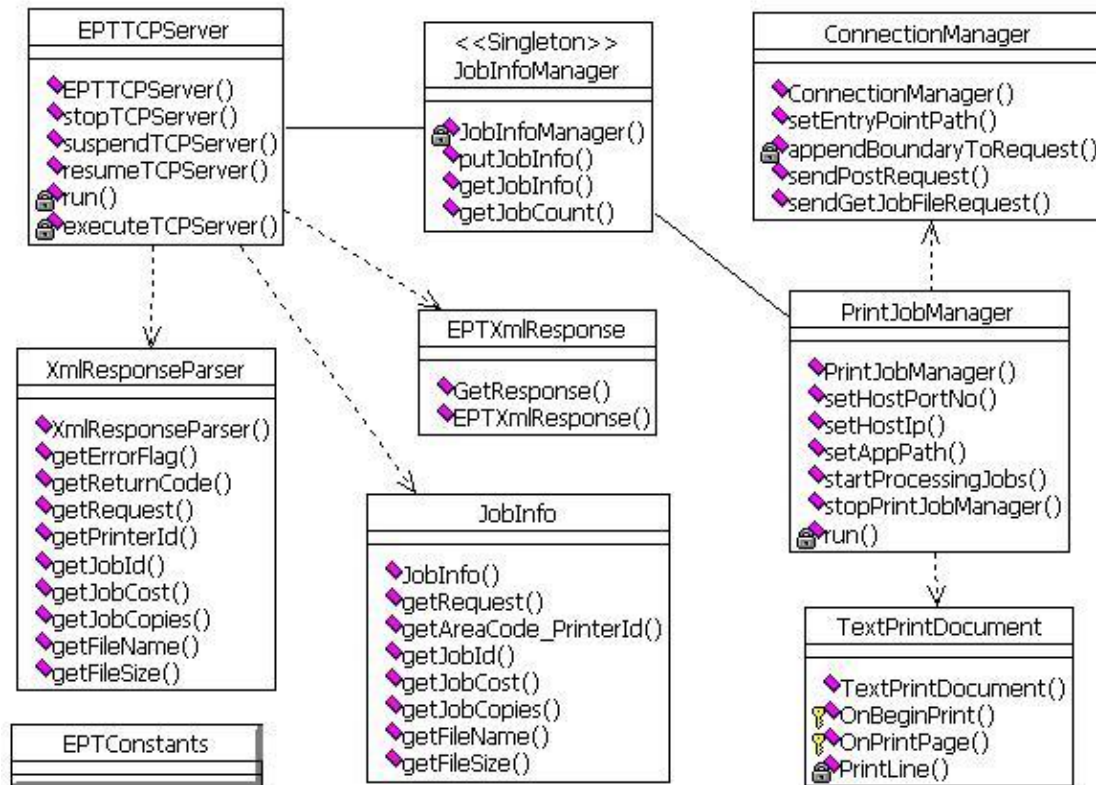
Figure 14: EPrinterTerminal Class Diagram

The following table describes the class name and its functionality:

Class Name	Functionality
PrinterCap	Print Capability data class
PaperInfo	Paper Information data class
EPTConstants	Holds the constants used by the all the classes.
EPTXmlRequest	This class consists of all the static methods to generate various requests
EPTXmlResponse	Gets the XML responses sent by the EAS.
ConnectionManager	Establishes connections to a server and transmits data.
XmlResponseParser	Parses the XML response sent by the EAS.
XmlFileReaderWriter	Used to read and write XML configuration files.
FrmEPrinterTerminal	The main controller class of the application.
FrmRegisterPrinter	Displays the Printer Registration form.
FrmRPRAddPaperInfo	Display the Paper Information form.
FrmEPTConfig	Displays the EPT configuration information form.
FrmAbout	Displays the About Information form.

Table 12. EprinterTerminal Classes - I

4.9.3 EPT Printing Process Module



EPrinterTerminal Printing Functionality Class

Figure 15: EPrinterTerminal Printing Class Diagram

Class Name	Functionality
EPTConstants	Holds the constants required by all the classes in the application.
EPTTCPServer	Controller class that binds to a port where the EPT application is running.
JobInfoManager	This is a singleton class, which holds the print job data objects.
ConnectionManager	Used to establish connects to the EAS server.
JobInfo	Job Information data object
XmlResponseParser	Parses the XML response.
EPTXmlResponse	Gets the XML response from the EAS.
PrintJobManager	This class prints the job file data onto the printer.
TextPrintDocument	This is a helper class to print text documents.

Table 13. EprinterTerminal Classes -II

This module is responsible for receiving the print requests from the EAS and ultimately to print the job on to the printer.

The printing process functionality is described as follows:

When the EPT starts, it starts the "EPTTCPServer" which binds to a port, listening for incoming print job requests from the EAS. When a job arrives the "EPTTCPServer" sends an acknowledgement to the EAS, and then sends the new job information into the "JobInfoManager". The "JobInfoManager" is a singleton class [Refer bibliography reference 6] i.e. there is one instance of this class, at any point of time during the execution of the application. The "JobInfoManager" notifies the "PrintJobManager" and wakes up the print job thread, this thread reads the new job information and sends a Get Job File (GJF) request to the EAS using the "ConnectionManager", if the job is successfully downloaded, it is then sent to the printer, which ultimately prints it. The print job thread then shifts to a sleep state.

5. Testing Process

This section describes the testing methodology; test plan and the list of test cases are also included.

5.1 Testing Methodology

5.1.1 Software Development Process

V-process model has been adopted for development. V-process model is Wipro's customized approach of the Waterfall model of software development.

5.1.2 Quality Process

The testing process comprised of

- ❖ Unit testing that tests the source code for syntactical & logical errors in each of the modules independently.
- ❖ Integration Testing that checks for the possible errors after the modules are integrated.
- ❖ Acceptance Testing tests the application on the real-time system at the client's site.

5.2 Test Plan

This section explains the test plan for testing the iPrint Service application and lists the test cases to test the various features in each module.

Features to be tested in the EAS system

Features
Printer Registration
Printer Deregistration
Printer status notification
Job status notification
Print Job Cancellation

Features to be tested in the Job Queue Controller

Features
Receiving Job Information
Sending Job Information to the EPT
Exception handling

Features to be tested in the Epson Printer Terminal

Features
Registering a printer
Deregistering a printer
Notifying the Job status
Notifying the Printer status
Receiving a New Job

5.2.1 Test Approach

- ❖ All features are to be tested as specified in the test plan
- ❖ The test cases present in each feature are to be executed sequentially.
- ❖ The error messages logged will be checked for their correctness.
- ❖ The entire test would be repeated after a bug is fixed.

5.2.2 Test Entry Criteria

The following sub-systems on the server should be in working condition

- ❖ Messaging Queue server.
- ❖ Database server.
- ❖ EPrinter Terminal.
- ❖ Web based application.
- ❖ Printer.
- ❖ Test data sets are ready.

5.2.3 Text Exit Criteria

The unit testing is considered complete if all features have been tested and there are no defects pending.

5.3 Test Cases

This section lists down the various test cases for the EAS system, Job Queue Controller and the EPT.

5.3.1 EAS System

To test the EAS system EPT Dummy must be used.

Test case ID	Test case description and steps	Expected Result	Remarks (Pass-P /Fail-F)
5.3.1.1	Send a valid RPR request to the EAS. Verify with the expected request.	Check the return code. RC = 200	Pass
5.3.1.2	Send any request with an invalid XML format to the EAS. Verify with the expected request.	RC = 300	Failed (Retested)
5.3.1.3	Send an invalid request type in any request. For instance send RXR instead of RPR. Verify with the expected request.	RC = 301	Pass
5.3.1.4	Send an invalid printer id in any request. For instance send XYZ_100 as printer id. Verify with the expected request.	RC = 302	Pass
5.3.1.5	Send invalid printer capability information in the RPR request. For instance set the color value to 20. Verify with the expected request.	RC = 303	Pass
5.3.1.6	Send invalid paper information in the RPR request. For instance the width and height have to be integer. Verify with the expected request.	RC = 304	Pass
5.3.1.7	Send an invalid Job Information. For instance send an invalid job status value in the NJS request. Verify with the expected request.	RC = 305	Pass
5.3.1.8	Send invalid file information in the GJF request. Verify with the expected request.	RC = 306	Pass
5.3.1.9	Send invalid printer status value, the printer status value cannot be other than 0 or 1. Verify with the expected request.	RC = 307	Failed (Retested)

Table 14. EAS System Test cases

5.3.2 Job Queue Controller

The correct operation of the Job Queue controller can be verified using the log files that are generated by the JQC.

Test case ID	Test case description and steps	Expected Result	Remarks (Pass-P /Fail-F)
5.3.2.1	The Job Queue controller should have three buttons Start, Stop and Exit.	3 Buttons should be present.	Failed (Retested)
5.3.2.2	When the start button is clicked. The stop	Identical	Failed

	button should be enabled and the start button should be disabled.	behavior as described.	(Retested)
5.3.2.3	When the Job Queue controller starts it should pick up all the messages from the queue.	Identical behavior as described. Check the log files to confirm.	Pass
5.3.2.4	Place a NWJ message in the queue and check the JQ. The NWJ message should be picked up and there should be no messages present in the JQ.	Identical behavior as described. Check the log files to confirm.	Pass
5.3.2.5	The NWJ message should be processed by the JQ and the database should be updated accordingly.	Identical behavior as described. Check the log files to confirm.	Pass
5.3.2.6	If the JQC is unable to transmit the job information to the EPT, it must update the database with the job status as failed/retry.	Identical behavior as described. Check the log files to confirm.	Pass
5.3.2.7	Start the EPT; the NWJ message, which was picked up by the controller, should be received by the EPT. This completes the flow the entire application.	Identical behavior. Check the log files to confirm.	Pass

Table 15. Job Queue Controller Test cases

5.3.3 EPT

Test case ID	Test case description and steps	Expected Result	Remarks (Pass-P /Fail-F)
5.3.3.1	Printer Registration. When the EPT is installed on a system. It has to be registered. Select the "Register Printer" menu item and click on it. A dialog to enter the registration information must be displayed. Fill up the dialog and press OK button. After the printer is registered successfully, the menu item must be	An ICON must appear in the status bar with the Printer Registration complete message. The database	Failed (Retested)

	disabled.	shall have a new entry in the printer table.	
5.3.3.2	Printer De-registration: The EPT can be de-registered when not in use. Select the "De-register Printer" menu item and click on it. The printer must be de-registered.	An ICON that appeared earlier must not be present in the status bar with the appropriate message. Check the database printer de-registration flag in the printer table.	Failed (Retested)
5.3.3.3	Job Information List: When a job arrives at the EPT for printing the Job Information List must be updated and job information must be listed.	Identical behavior as described.	Pass
5.3.3.4	Clearing the job information list. Right clicking on the job information list and selecting delete in the menu should clear the Job information list for the selected job.	Identical behavior as described.	Pass
5.3.3.5	Changing the EPT configuration information. This information tells EPT which EAS system to connect to. This must be done before registering the printer. Click on the menu item "Configuration". The server name and the port no must be specified. This information must be retained even after a restart.	Identical behavior as described.	Pass
5.3.3.6	About Information Select the menu item "About"; this information describes the version of the software. Clicking on the OK button should close the dialog.	Identical behavior as described.	Pass
5.3.3.7	Notifying the printer status. EPT sends a NPS request as soon as it starts.	Check the log file.	Pass
5.3.3.8	Notifying the Job status. EPT sends a NJS request as soon as a job gets printed.	Check the log file.	Pass

Table 16. EPT Test cases

6. Deployment Environment Considerations

This section explains the development, deployment environments and explains the deployment using the deployment diagram.

6.1 Development Environment

Category	Technologies/IDE/Platform
Language	C#
Server Side	ASP.NET
Messaging Middleware	Microsoft Messaging Queues (MSMQ)
SDK	Microsoft Mobile Internet Toolkit, .NET Framework
Web Server	Internet Information Server 5.0 (IIS)
Database Server	SQL Server 2000
IDE	Visual Studio.NET (VS.NET)
Operating System	Windows 2000 Server

Table 17. Development Environment

6.2 Deployment Environment

Category	Technologies/ Platform
Messaging Middleware	Microsoft Messaging Queues (MSMQ)
Runtime	. NET Runtime
SDK	Microsoft Mobile Internet Toolkit, .NET Framework
Web Server	Internet Information Server 5.0 (IIS)
Database Server	SQL Server 2000
Operating System	Windows 2000 Server

Table 18. Deployment Environment

6.3 Deployment Diagram

In order for this system, to run at optimal performance the following deployment scenario has to be considered:

The whole application can be divided into two parts:

- ❖ Client Deployment
- ❖ Server Deployment

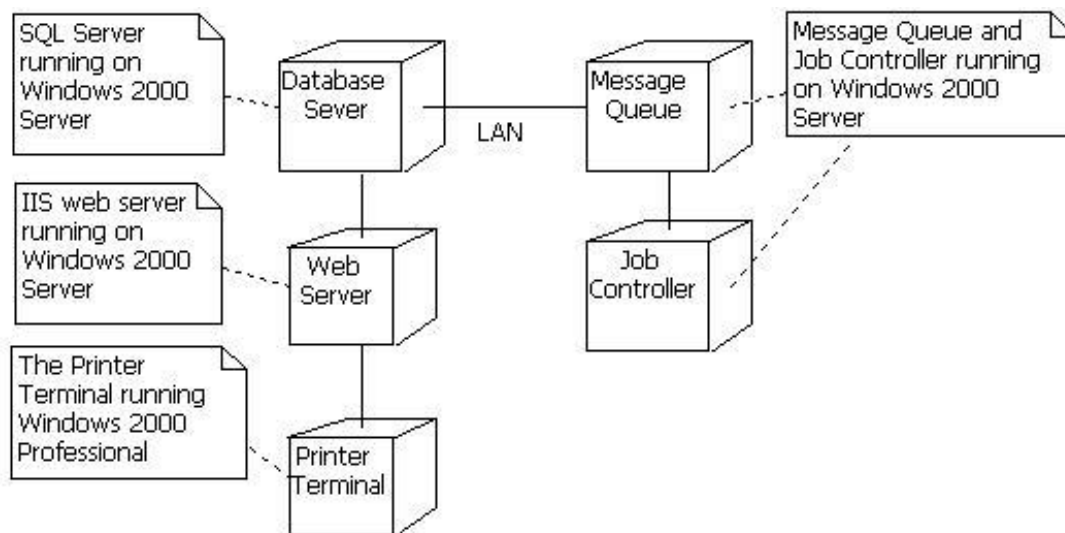


Figure 16: Deployment Diagram

6.4.1 Server Deployment

The server application would consist of different machines with one server running on each machine.

- ❖ Database Server (SQL Server 2000)
- ❖ Message Queue system (MSMQ Server) and Job Controller
- ❖ Web Server (IIS 5.0)

6.4.2 Client Deployment

The client would be ultimately a printer on top of which the "EprinterTerminal" application would be running.

For testing purposes it would be a PC with a printer attached to it.

7. Conclusions and Deductions

- ❖ The entire application is minimally stateful, which makes it very scalable. Any system, which has minimal state, is more scalable.
- ❖ The avoidance of polling mechanism increased the scalability of the system.
- ❖ Response time of the EAS system significantly depends upon the response time of the sub systems (messaging queue, controller, web server and database system).
- ❖ The actual cell phone cannot be used to test the application. A cell phone simulator shall be used to test the application.
- ❖ The EPT shall be finally a printer with the required capabilities, for the purpose of “proof of concept”, it is an application that runs on a PC connected to a printer.
- ❖ The areas like user subscription, maintaining user profiles, user preferences, logging in to the service and billing are not included in the current scope of work.

8. Recommendations

8.1 Scalability

In this architecture all the sub-systems are loosely coupled and hence each sub-system can be independently scaled. Adding more number of servers to the system will scale out the EAS.

Message Queuing sub system can be scaled by adding a new Message Queuing server to the iPrint Service.

8.2 Load Balancing

A load balancer can be used to balance the load of the incoming requests. Thus the load on the EAS system will be balanced. Load balancers support various ways of load balancing like Round robin, no of parallel/simultaneous connections etc. Administrators can configure the method of load balancing.

The load on the Message Queue Server will be handled by JQC. The design of JQC is involves parallel processing, it spawns a thread to process every message in the message queue.

8.3 Availability

The possible points of failure are the EAS System, Job Queue controller and common utilities like Load Balancer, Message Queuing System, and Database Server.

The functionality of EAS is replicated on other servers also. This ensures availability even if any of the EAS Servers fail. Load balancer automatically detects the failed server and stops dispatching the requests to that server.

In case any of the Job Queue Controller servers fail, the other servers having same functional responsibility will take over the load.

The Load Balancer provides built-in hot standby and fail-over support. For fail-over support to the database, data mirroring feature provided by the database can be used. For Message Queuing System we need to have a standby Message Queue Server.

8.4 Administration of servers

For administration & monitoring the health of the servers in the Server Farm, external tool will be needed. In case of Application Server the in built support is provided to monitor the servers in the cluster. Apart from these the web-based interface will be provided to administer the application specific information.

If the site is hosted then Data Center also provides the support for monitoring and reporting of load on the servers.

9. Appendix

9.1 EAS SQL Installation script

```
-- Install script begins here...

-- Uses the EAS database
use EAS

-- Drops the tables if already existing
drop table paperinfo
drop table jobinfo
drop table printer

-- Creates the printer table
create table printer
(
    printer_guid uniqueidentifier DEFAULT NEWID(),
    area_code varchar(20),
    printer_id varchar(20),
    printer_status char(1) default '0' not null,
    printer_status_date datetime DEFAULT (getdate()),
    manufacturer varchar(30),
    model varchar(30) not null,
    type char(1) not null,
    color char(1) not null,
    duplex char(1) not null,
    deregister varchar(5) default 'false' not null,
    creation_date datetime DEFAULT (getdate()),
    deregister_date datetime DEFAULT (getdate()),
    ept_host_ip char(15) not null,
    ept_host_portno char(5) not null,
    constraint PK_PRINTER primary key (area_code, printer_id)
)

-- Creates the Paper Information table
create table paperinfo
(
    area_code varchar(20),
    printer_id varchar(20),
    paper_size varchar(10),
    paper_width varchar(7) default '0',
    paper_height varchar(7) default '0',
    constraint PK_PAPERINFO PRIMARY KEY (area_code, printer_id,
paper_size),
    constraint FK_PAPERINFO FOREIGN KEY (area_code, printer_id)
REFERENCES printer (area_code, printer_id)
)
```

```

-- Creates the job information table

create table jobinfo
(
    area_code varchar(20),
    printer_id varchar(20),
    job_id varchar(20),
    job_status char(1) default '2' not null,
    job_filename varchar(255) not null,
    job_filesize numeric(10) not null,
    copies tinyint not null,
    cost numeric(7,2) not null,
    job_date datetime DEFAULT (getdate()),
    constraint PK_JOBINFO primary key (job_id) ,
    constraint FK_JOBINFO FOREIGN KEY (area_code, printer_id)
    REFERENCES printer (area_code, printer_id)
)

-- For every area code registered, there needs to be at least one entry
-- per area code.

-- WIPRO area code
insert printer (area_code, printer_id, manufacturer, model, type,
color, duplex, ept_host_ip, ept_host_portno, deregister)
values ('WIPRO', '100', 'DUMMY', 'DUMMY', '2', '1',
'1','localhost','8080', 'true')

-- MDP Area code
insert printer (area_code, printer_id, manufacturer, model, type,
color, duplex, ept_host_ip, ept_host_portno, deregister)
values ('MDP', '100', 'DUMMY', 'DUMMY', '2', '1',
'1','localhost','8080', 'true')

-- PUN area code
insert printer (area_code, printer_id, manufacturer, model, type,
color, duplex, ept_host_ip, ept_host_portno, deregister)
values ('PUN', '100', 'DUMMY', 'DUMMY', '2', '1',
'1','localhost','8080', 'true')

-- Install script ends here

```


9.2 A Message Queuing Analogy

Comparing telephone conversations and electronic mail exchanges provides a good analogy. With a telephone conversation, callers can exchange information immediately and quickly. The telephone is much less useful, however, when one of the two parties cannot be contacted. The telephone line may have been severed, or the other party may be busy doing other tasks or simply out to lunch. In any case, the calling party must continue to dial the receiving party periodically in order to deliver an important message.

In the case of electronic mail, on the other hand, the calling party could simply send an e-mail message and move on to other tasks, knowing that the receiver will eventually get the message and act appropriately. Of course, what makes e-mail a viable alternative to the telephone is the knowledge that messages will be delivered reliably and receivers will eventually read their mail and perform required actions. Message queuing is like electronic mail, except senders and receivers are application programs instead of people—and messages are data instead of electronic letters.

Eleven guidelines for writing better MSMQ applications

Quick summaries of the programming guidelines for MSMQ are mentioned in this section. One should follow them when writing an MSMQ application.

- Do only local receives.
- Avoid functions that query the Message Queue Information Store (MQIS).
- Implement timeouts.
- Understand the limits of asynchronous notification.
- Know when and where to use transactions.
- Know when to use persistable COM objects.
- Understand what security context to use.
- Implement smart queue usage.
- Request acknowledgements or no acknowledgements.
- Remember case sensitivity.
- Test your application with a full reboot while offline.

10. References/Bibliography

1. Microsoft Developers Network (MSDN) - <http://msdn.microsoft.com>
2. Active Server Pages Quick Start Tutorials –
<http://www.asp.net>
<http://www.dotnetjunkies.com/quickstart/aspplus/>
3. Microsoft Messaging Queue White Paper (MSMQ) –
http://www.microsoft.com/msmq/MSMQ3.0_whitepaper_draft.doc
4. Defining the Basic Elements of .NET -
<http://www.microsoft.com/net/defined/whatis.asp>
5. Web and Application Services -
<http://www.microsoft.com/windows2000/technologies/web/default.asp>
6. Design Patterns Explained (*A new perspective on Object-Oriented Design*) by Alan Shalloway, James R. Trott, Addison-Wesley.
7. Microsoft SQL Server 2000 –
<http://www.microsoft.com/sql/default.asp>
<http://msdn.microsoft.com/vstudio/techinfo/articles/servers/radserver.asp>
8. HTTP 1.1 RFC 2616 - <http://www.faqs.org>
9. Beginning XML – by David Hunter, Wrox Press.

Check list of items for final report

- | | |
|---|-------|
| a. Is the Cover page in proper format? | Y / N |
| b. Is the Title page in proper format? | Y / N |
| c. Is the Certificate from the Supervisor in proper format? Has it been signed? | Y / N |
| d. Is Abstract included in the Report? Is it properly written? | Y / N |
| e. Does the Table of Contents' page include chapter page numbers? | Y / N |
| f. Is Introduction included in the report? Is it properly written? | Y / N |
| g. Does the Report contain a summary of the literature survey? | Y / N |
| i. Are the Pages numbered properly? | Y / N |
| ii. Are the Figures numbered properly? | Y / N |
| iii. Are the Tables numbered properly? | Y / N |
| iv. Are the Captions for the Figures and Tables proper? | Y / N |
| v. Are the Appendices numbered? | Y / N |
| h. Does the Report have Conclusions / Recommendations of the work? | Y / N |
| i. Are References/Bibliography given in the Report? | Y / N |
| j. Have the References been cited in the Report? | Y / N |
| k. Is the citation of References / Bibliography in proper format? | Y / N |