

# **Arch WCTP Proxy Server User Manual**

## **Servlet Edition**

*Revision: 0.0.1 Alpha Document*

## Table of Contents

- 1 Overview ..... 3
  - 1.1 Client Messages and Enterprise Requests..... 3
  - 1.2 Client Queries..... 3
  - 1.3 Pushed Receipts..... 4
  - 1.4 Database Operations..... 4
    - 1.4.1 Inserting a Receipt..... 5
    - 1.4.2 Querying a Receipt..... 5
    - 1.4.3 Updating a Receipt..... 5
    - 1.4.4 Purging the Database..... 5
- 2 Installation Instructions ..... 6
  - 2.1 Create the database and database tables..... 6
  - 2.2 Set up the Properties Files..... 7
    - 2.2.1 receivelet.properties ..... 7
    - 2.2.2 db.properties..... 7
  - 2.3 Build and Install the WAR File..... 7
- A Glossary of Terms ..... 8

## List of Figures

- Figure 1-1 Client Message to Enterprise Request..... 3
- Figure 1-2 Client Queries..... 3
- Figure 1-3 Proxy Receiving Pushed Responses..... 4

# 1 Overview

## 1.1 Client Messages and Enterprise Requests

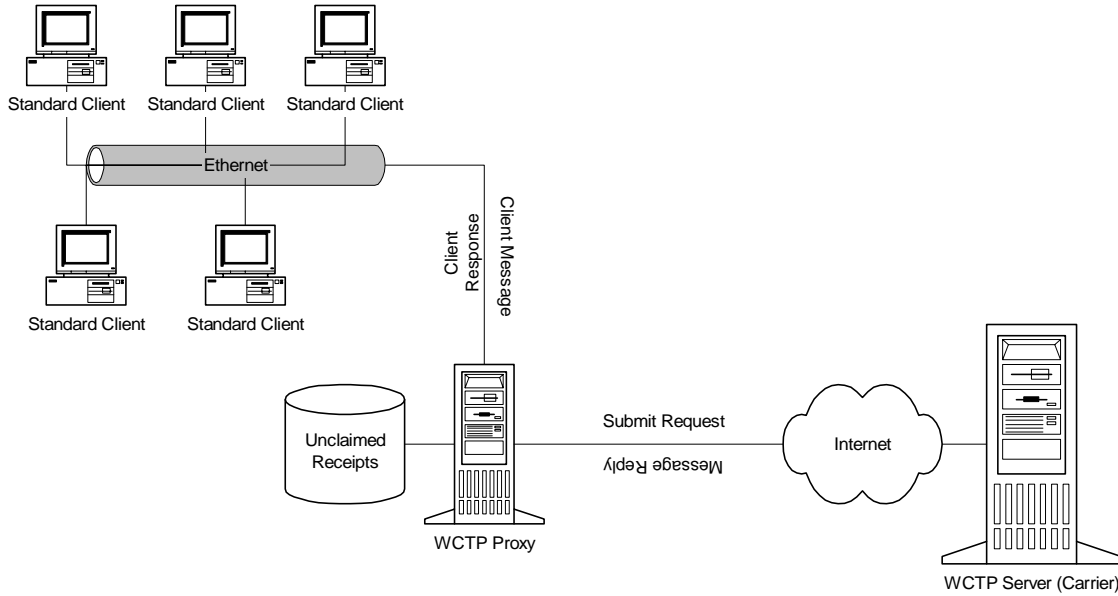


Figure 1-1 Client Message to Enterprise Request

The proxy receives a message from a transient client and creates a wctp-SubmitRequest packet that is sent to the WCTP Server. The senderID and messageID fields are transposed from the wctp-ClientMessage packet to the wctp-SubmitRequest packet. The WCTP Server responds with a wctp-MessageReply packet. The proxy receives the reply and generates the appropriate message to the transient client. Figure 1-1 shows transient client on a LAN generating wctp-ClientMessage packets to a proxy. See section 1.4.1 for more information regarding the database operation.

## 1.2 Client Queries

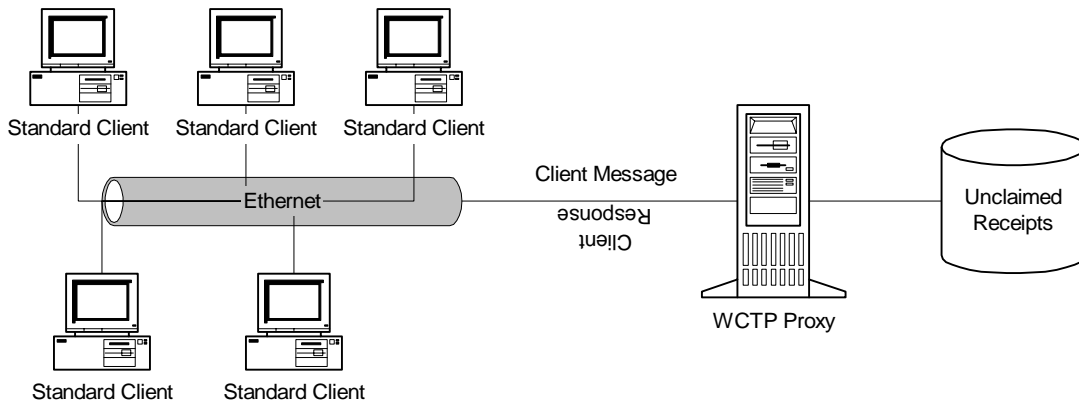
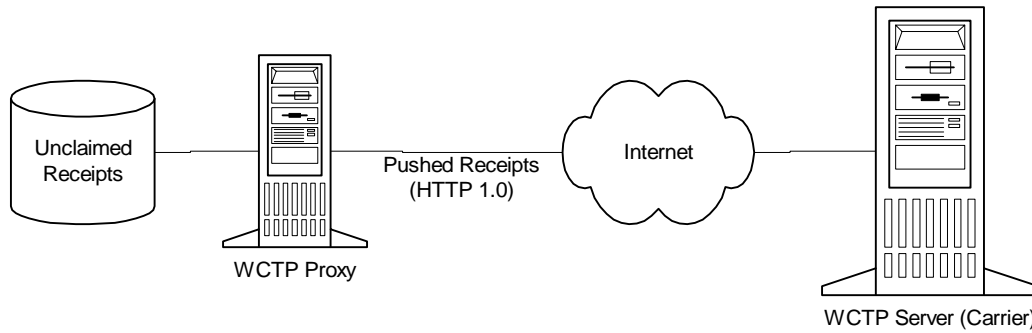


Figure 1-2 Client Queries

A transient client queries the proxy using the wctp-ClientQuery operation. The proxy checks its database and responds with a wctp-ClientQueryResponse packet. The proxy stores all client query requests as undelivered receipts until a receipt has been provided or until it expires due to time limits. Figure 1-3 shows transient client on a LAN generating wctp-ClientQuery packets to a proxy.

### 1.3 Pushed Receipts



**Figure 1-3 Proxy Receiving Pushed Responses**

The preferred model for a proxy to obtain receipts from a WCTP Server is to have the receipts pushed back to the proxy from the WCTP Server. The proxy simply listens on a port for receipts being posted to that port using the HTTP protocol. The proxy updates records in the database with delivery confirmations at the time when receipts are received.

### 1.4 Database Operations

Database wctp\_receipts

**receipts\_mst**

PK			FK		
<i>char(50)</i>	<i>char(50)</i>	<i>char(256)</i>	<i>int</i>	<i>char(256)</i>	<i>Char(19)</i>
track_no	message_id	sender_id	notification	recipient_id	timestamp
1234.23xz	1234.23xzArch	randy@arch.com	1	1234567	2000-03-1T19:45:00
1234.239z	1234.239zArch	joe@arch.com	2	7654321	2000-03-1T19:45:10
1234.982z	1234.982zArch	sue@arch.com	3	5545545	2000-03-1T19:46:22

**notification\_mst**

PK	
<i>int</i>	<i>char(10)</i>
type	descr
1	Queued
2	Delivered
3	Read
4	Unknown

**Table 1: Database Definition**

### 1.4.1 Inserting a Receipt

At the time that the proxy receives a response from the WCTP server the proxy detects if the transient client has requested a receipt and, if so, inserts a new record into the database. A unique key is generated and that is duplicated and appended with the enterprise's name. (See section 2.2.1 to see how to set this in the properties file) The initial notification type is set as "Unknown."

If the response was a failure, an entry is not made in the database. After that point, the client will receive a failure or success response from the proxy based on the response from the server.

### 1.4.2 Querying a Receipt

A receipt is queried in the database when a transient client performs a wctp-ClientQuery. See section 1.2 for more information.

### 1.4.3 Updating a Receipt

A receipt is updated in the database when new status updates are received from the WCTP server. This happens when a wctp-StatusInfo operation is pushed to the proxy servlet. See section 1.3 for more information.

### 1.4.4 Purging the Database

The database can be purged at the discretion of the proxy administrator. The carrier, however, is likely to have a predefined set of rules on how long they will continue to try to deliver a message. For example, the carrier may give up on a delivery confirmation after 72 hours. If the proxy does not purge the database periodically, the database could experience problems due to an unexpectedly fast growing table.

Different operating systems and database engines have different ways to periodically purge a database. For example, a UNIX System Administrator might set up a cron job to periodically run a program that purges the database.

This release does not provide a way to purge the database through the Servlet.

## 2 Installation Instructions

The WCTP proxy server has been successfully installed and implemented on a Linux server running the Tomcat Servlet engine and a PostgreSQL database server. Other Servlet engines that are compliant with the Sun specification should also work. Any database that has a JDBC connector should also work. Only standard JDBC features are used in the database connectivity.

### 2.1 Create the database and database tables

First, a database with the name of “wctp\_receipts” must be created. Next create the tables using the structure shown in Table 1. The following SQL can also be used as a reference point:

```
CREATE TABLE receipts_mst (  
  track_no char(50) NOT NULL,  
  message_id char(50) NOT NULL,  
  sender_id char(25) NOT NULL,  
  notification INTEGER NOT NULL,  
  recipient_id char(256) NOT NULL,  
  timestamp char(19) NOT NULL,  
  PRIMARY KEY (track_no)  
);
```

```
CREATE TABLE notification_mst (  
  type INTEGER NOT NULL,  
  descr char(10) NOT NULL,  
  PRIMARY KEY (type)  
);
```

Insert the notification types and descriptions into the notification\_mst table:

```
INSERT INTO notification_mst (type, descr)  
VALUES (1, 'QUEUED');  
  
INSERT INTO notification_mst (type, descr)  
VALUES (2, 'DELIVERED');  
  
INSERT INTO notification_mst (type, descr)  
VALUES (3, 'READ');  
  
INSERT INTO notification_mst (type, descr)  
VALUES (4, 'UNKNOWN');
```

Now install the JDBC driver specific to your database. This step is implementation specific, however, many Servlet engines will allow you to drop the jar file into a lib directory that serves all Servlets. If you are only using database access for this single

application, you may want to build the war file with the addition of the jar file into the WEB-INF/lib directory.

## 2.2 Set up the Properties Files

Two properties files need to be setup, receivelet.properties and db.properties. Both files are located in the WEB-INF directory.

### 2.2.1 receivelet.properties

carrier	The URL of the gateway you are using
enterpriseID	What you would like to use as the senderID field to send to the WCTP server. This normally instructs the gateway where to post receipt statuses too.
companyName	This helps to build a unique messageID to send to the WCTP server. Not required, but recommended.

**Table 2: Receivelet Properties File**

*Example:*

```
carrier=http://www.arch.com:80/WCTP
enterpriseID=MSG:someuser@www.yourserver.com/wctp/wctppost/
companyName=Arch
```

### 2.2.2 db.properties

db_driver	The fully qualified name of your JDBC database driver.
db_URL	The JDBC URL of your database.
db_user	Your username for the database server.
db_password	Your password for the database server.

**Table 3: Database Properties File**

*Example:*

```
db_driver=org.postgresql.Driver
db_URL=jdbc:postgresql://localhost/wctp_receipts
db_user=username
db_password=password
```

## 2.3 Build and Install the WAR File

First, check to make sure none of the included files interfere with your installation. For example, if you already have JDOM or Xerces available on your system, there is no need to leave them in the WEB-INF/lib directory. Next, add a JDBC driver to the WEB-INF/lib directory. You also may need to edit the web.xml file to suite your specific implementation. Installing the WAR file is implementation specific.

## A Glossary of Terms

**Proxy** – A proxy acts as an intermediary between a client and a server. In this case, the proxy is used to store receipt information and forward WCTP messages.

**Server** – The term is used throughout this document in the sense that a server exists to talk to a proxy. Clients talk to the server through the proxy. This allows an enterprise greater control of messages that are being sent to a wireless network. A server would most commonly be a carrier.

**WCTP (Wireless Communications Transfer Protocol)** – A protocol specification maintained by the PCIA that is designed to deliver messages to wireless devices and applications.

**XML (Extensible Markup Language)** – Specification maintained by the W3C (World Wide Web Consortium.)