Powerfares will work on Windows 8 and Windows server 2012 only. Minimum 4GB memory and minimum 256GB disk (either SSD or mechanical). For fast-mode A2A, minimum 8GB memory.

# Introduction

Powerfares is a minimal webserver with the addition of JSON-based APIs to provide all (“anywhere to anywhere”) UK rail fares and reference data to support encoding of CCST magnetic stripes and CCST print formats. Point to point timetable information is provided for direct trains. It can also serve any static web page but does not provide server-side scripting nor does it support "*transfer-encoding chunked*" (which is designed to allow dynamic server side page generation).

The ability to serve web-pages means that browser based fares query applications can be developed on a client device (mobile or TVM-based). However, any application (web, C# or otherwise) can query the JSON APIs for fares and timetable information.

When running on a server, powerfares supports connections efficiently from hundreds of clients typically processing a fares request in a few hundred microseconds. It has full multithreading support and will typically run one thread per CPU (or CPU core) and queue requests to threads as they become free. It is thus capable of providing a fares information service to hundreds of TVMs.

When running on a TVM, exactly the same capabilities are available but they are subject to the amount of memory and the capabilities of the TVM’s CPU. For example, a two-core Intel Atom D525 is unlikely to be capable of handling thousands of fares requests per second but it would be conceivable to use a TVM to serve fares to other local devices in the same station (perhaps a very thin-client TVM just running a browser).

However, most of the time it is envisaged that if an instance of Powerfares is running on a TVM that it would be providing fares only for the TVM.

# Fares and timetable server functions

The product serves all UK rail fares, Plusbus fares and gives correct prices for adult, child and railcard discounts. It provides a full reference data service so that all data required for the magnetic stripe and ticket printed is included. It provides timetable information for direct trains only.

# Modes of operation

The product works in two modes “server mode” which is appropriate for multiple clients (such as a network of hundreds of TVMs) and “embedded mode” which is appropriate for use in a TVM with low memory (minimum 4GB).

The only difference between the two modes is that in server mode, the entire RJIS data set is read into memory, but in embedded mode, Powerfares expects to import fares data from “per-origin” pre-filtered RJIS files at the time when a secondary origin is chosen by the end customer (the primary origin is the TVM’s “home” origin – the station where it is installed). Per-origin fares are produced by the ***RJISPrefilter*** tool and allow the product to serve all UK rail fares in a low-memory (4GB) environment. All other features (such as the ability to serve web pages) are unaffected by the choice of modes though the ability to serve multiple clients is limited when running in embedded mode is limited – a recommended maximum would be ten clients.

While running in an embedded environment (such as a TVM) Powerfares is still capable of serving web pages to any connecting client. It can thus be used to provide basic statistics or diagnostics for the TVM.

# Web server.

The product serves JSON over http – its core APIs are accessed by sending http GET requests to one of several “well-known” URLs. The product serves files over http (in the same way any webserver does) but also provides a method to map a specific URL to an API in the configuration file **pfconfig.xml**. APIs have hardcoded names: FARES, REFDATA, CACHEHINT etc. and are mapped to URLs in pfconfig.xml. For example, a statement like

<apimap name=”FARES” suffix=”GETFARES”>

means that a the URL with the GETFARES suffix will be mapped to the fares API. A query for fares would then look like this (in a browser running on the same device as powerfares)

http://127.0.0.1/FARES?o=5883&d=3115&r=YNG&d0=2015-11-30&d1=2016-01-15

This is a request for all fares with a young person’s railcard from Poole to Oxford on the 30th November 2015 returning on the 15th January 2016.

# Datafeed import

Powerfares requires the following data feeds:

• RJIS (RSPS 5045)

• IDMS (RSPS 5044)

• Timetable (RSPS 5046)

• A future enhancement will read .plan files in order to perform journey planning

Fares are imported on start-up directly from the original RJIS files (RSPS 5045) and timetable information is imported directly from the RSP timetable datafeed files (RSPS 5046). The product running in server mode takes around 15 seconds to read the complete set of fares and full timetable information on a powerful (Intel I7) PC or Xeon server with 16MB RAM. It is anticipated to take around 30-45 seconds on an Intel NUC with 8GB memory. In embedded mode running on an Intel Atom D525 the product will take around 10 seconds to read a pre-filtered fare set on startup (a fare set for the TVM’s “home” station) then a few hundred milliseconds to read fares for a different origin each time a new origin is selected by the end customer.

# RJIS prefilter tool

A filtering tool is necessary to enable anywhere to anywhere functionality on “low-memory” (i.e. 4GB) tvms. This tool will separate fares by origin into around 2500 fares sets – one for each station in the UK. Each time the passenger selects a different origin at the TVM, a set of fares for that station is loaded – this will be loaded in the background by the Powerfares program so that by the time the passenger selects a destination the complete fare set for that origin is loaded. Powerfares will provide a “cachehint” API which loads the fareset for a certain origin into memory. The loaded fare set remains in memory until a new origin is requested. Note that the fares for the TVM’s “home” station are loaded into memory permanently.

A rudimentary filtering program to produce filtered fares sets for all stations in the UK already exists, however, it was designed for filtering fares by TOC and as such would be slow to produce an individual fares set for each station. Currently it is run once to produce a TOC-specific filtered fares set, but to produce fares set for 100 stations it would need to be run 100 times taking around two hours).

# Exclusions

## Operating systems

Powerfares will never work on operating systems below Windows 8 or server operating systems below windows server 2012.

## Low memory systems

Powerfares will never work on systems with below 4GB of total RAM.

## PMS

Powerfares will not read the new PMS feeds from RJIS. The content of this feed is not yet defined and the feed is not ready for testing. However, it is known that the PMS feed will be XML which is radically different from the existing RJIS format. It is thus expected that significant effort will be required to import this data and data important times may be significantly longer.

## Automatic fares update.

Powerfares does not support automatic update of fares by any “push” technology. Fares have to be manually placed on the server or TVM, or the TVM has to use its own download technology to acquire fares data. A restart of Powerfares will always cause it to re-read the entire fare set.

## Journey planning

The initial version does not include journey planning, however there is a prototype architecture to achieve journey planning.

## Live train information

The initial version does not include the Darwin live feed, however this could be added.

## Reservations and booking

Powerfares is an information-only product. It does not support any mechanism for booking, reservation or settlement. These features need to be provided by other software services in the TVM or in the Parkeon back office system.

# Future Enhancements

## Journey planning

This can be achieved by producing journey plans “offline” meaning that they are generated from timetable information and the RSP routeing guide data feed (RSPS 5047) on a system not necessarily connected to the TOC’s TVM network at the time. A set of journey plan files is produced – one for each station in the UK. Each .plan file contains one line for every other station in the UK which can be reached by an INDIRECT train journey. Each line contains a number of journey plans with each plan separated by a vertical bar characted. Each plan consists of a number of train change points.