

Deploying Linked Data

This document describes the process of deploying Linked Data into the existing Web. It discusses some of the difficulties faced in exposing RDF data and in bridging the "Linked Data Web" and the traditional "Document Web". Two generic approaches to resolving these deployment challenges are described, content negotiation and URL rewriting, before looking at [OpenLink Virtuoso](#), both from the standpoint of how it implements these solutions and how Linked Data is deployed.

A companion document, *Virtuoso RDF Views - Getting Started Guide*, focuses on Virtuoso RDF Views, a facility for exposing relational data as RDF. In addition, it provides useful background information for readers unfamiliar with RDF and outlines some of the key technologies of the Linked Data Web.

Contents

- Introduction ([Part 1](#))
 - What's The Current Problem?
 - What is Linked Data?
 - What Does the Linked Data Meme Enable?
 - Deployment Challenges
 - Exploiting HTTP URI Duality
 - Real-World Object Referencing
 - Resource Identity
 - Hash & Slash based URIs
 - Hash URIs
 - Slash URIs
 - Resolution of the Deployment Challenge
 - Content Negotiation
 - HttpRange-14 Recommendations
 - Content Negotiation & Linked Data Deployment
 - Content Negotiation with Hash URI Based Data Object Naming Scheme
 - Example 1: Static descriptions of a Data Object
 - Example 2: Dynamically derived descriptions of a Data Object using a SPARQL DESCRIBE
 - Content Negotiation with a Slash URI Based Data Object Naming Scheme
 - URL Rewriting
 - Using URL Rewriting to Solve Linked Data Deployment Challenges
- Deploying Linked Data using Virtuoso ([Part 2a](#))
 - The Virtuoso Rules-Based URL Rewriter
 - Conductor UI for the URL Rewriter
 - Virtual Domains (Hosts) & Directories
 - 'Nice' URLs vs. 'Long' URLs
 - Rule Processing Mechanics
 - Enabling URL Rewriting via the Virtuoso Conductor UI
 - Northwind Demonstration Database

- Configuring Rewrite Rules using Conductor
 - Dissection of Northwind Rewrite Rules Configured using Conductor
 - Regex Rule for RDF Requests
 - Constructing the Destination Path Format
 - Data Flow in Conductor-Defined Northwind RDF Regex Rule
 - Regex Rule for HTML Requests
- Enabling URL Rewriting via Virtuoso PL
 - Exporting Rewriting Rules from Conductor
 - Defining Virtual Hosts in Virtuoso PL
 - URL Rewriting Configuration API
 - Creating Rewriting Rules
 - URLREWRITE_CREATE_REGEX_RULE
 - Dissection of Northwind Rewrite Rules Configured using Virtuoso PL
 - Data Flow in Virtuoso/PL-Defined Northwind RDF Regex Rule
- Northwind URL Rewriting Verification Using `curl`
- Browsing & Exploring the Northwind RDF View ([Part 2b](#))
 - Interacting with Linked Data via ODE
 - Interacting with Linked Data via iSPARQL
- Transparent Content Negotiation ([Part 2c](#))
 - HTTP/1.1 Content Negotiation
 - Transparent Content Negotiation
 - Deficiencies of HTTP/1.1 Server-Driven Negotiation
 - Variant Selection By User Agent
 - Variant Selection By Server
 - Variant Selection By End-User
- Transparent Content Negotiation in Virtuoso HTTP Server
 - Describing Resource Variants
 - HTTP_VARIANT_MAP Table
 - Configuration using Virtuoso/PL
 - Variant Selection Algorithm
- Transparent Content Negotiation Examples
 - Simple TCN with Static Content
 - Northwind RDF View
 - DBpedia
- Simplifying Deployment with RDFa
 - No Content Negotiation or 303 Redirects
 - Generating RDFa Dynamically Using `Description.vsp`
 - RDFa Output From Non-RDF Data Sources
 - Sample RDFa Output From `Description.vsp`
- [Glossary](#)
- [Bibliography](#)
- [Appendix A: Description.vsp - Rendering RDF as HTML](#)
- [Appendix B: New Proxy URI Formats](#)