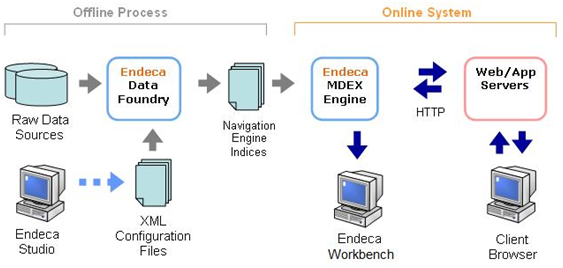
# Overview

**Project Overview**PBS has deployed the Endeca IAP solution that provides search, Guided Navigation, and dynamic merchandising functionality on the pbs.org site.PBS engaged with Thanx Media Professional Services for guidance Finish(of the Data Foundry and pipeline processes for the creation of the indices that is utilized by the sites). The user interface portion and control scripts for the application were developed by PBS to interact with the MDEX Engine through the Endeca Assembler API. **About the Technical Specification Document**The purpose of this document is to outline the implementation details for the Endeca IAP instance reflecting the requirements defined through a Requirements Workshop and subsequently captured in the PBS Business Requirements document. The Technical Specification document contains information about Endeca, index creation, custom scripts, and operational processes through control scripts and the Endeca Workbench.

# Overview of Endeca Technology

**Endeca Architecture Overview**The basic Endeca Architecture consists of four (4) main parts as illustrated in the diagram: the Endeca Data Foundry (EDF), the Endeca MDEX Engine (MDEX), the Endeca Application Controller (EAC), and the Endeca API.



**Endeca Data Foundry (EDF)**The EDF is an offline process that first reads in the source data and application specific configuration files. Using the instructions outlined in the configuration files; the EDF transforms the raw data into structured data and indexes it. The output of the EDF is a set of indices (binary and xml files) which are provided to the Endeca MDEX Engines. This is typically a scheduled offline process that runs, and therefore happens independently of the MDEX Engines handling the runtime queries. **Endeca MDEX Engine (MDEX)**The MDEX is an online server that listens on a specific port for HTTP requests, and responds by returning binary results. Once a new set of indices is created from the EDF, the MDEX will load the indices into memory at process restart. The MDEX is also stateless, with no concept of a session object for each user. This allows multiple MDEX engines with the same set of indices to return the same results, allowing for easy scalability and redundancy behind a load balancer. **Endeca Application Controller**The Endeca Application Controller consists of the Central Server and Agents. The EAC is installed and runs on the primary Endeca server, creating a single interface for coordinating jobs across all machines. Agents are installed on each Endeca server and are responsible for the actual work of an Endeca implementation. Communication to these agents is performed through the EAC Central Server via eaccmd, a command line interface, or Workbench. **Endeca API**The Endeca Presentation Layer Java API can be installed on any application server that supports Java. The API is used to create and send the query to the MDEX Engine over HTTP. It receives the response as a binary encoded result object and uses the appropriate methods and objects to display the information in the UI. The query must be submitted to a pre-defined host and port. If only one MDEX engine exists, the host and port of the MDEX engine should be configured at the UI level. If multiple MDEX engines exist behind a load balancer, the host and port of the load balancer should be configured at the UI level.  **Implementation Architecture**Architecture of the Endeca cluster will follow the Oracle Endeca best practice of Authoring/Live clusters. Authoring environments will be run with Live data, and will serve as the Endeca Workbench test platform. An overview of the environment can be found in section 16.

# Data Definition

**Data Sources**There are (4) main data types that make up the pbs.org site indices:

* Records Feed: Provides information that will be used for search, search results, and navigational purposes throughout the site.
* PS Record Data Feed: Provides program information that will also be used for search, search results, and navigational purposes throughout the site.
* Forum Data Feed: Provides the category information that will be used to display category information and setup the Product Category hierarchy for navigation.
* Search Rank Data Feed - Provides the category that will be used to display category information and setup the Product Category hierarchy for navigation.

**Crawl Data**

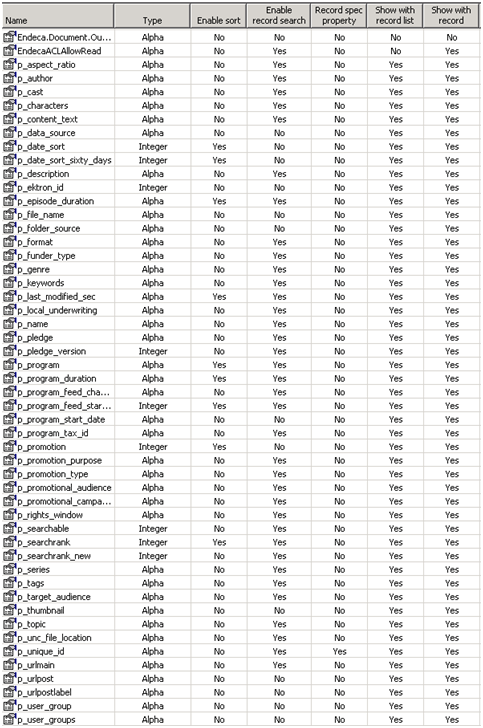
**Partial Pipeline**There are (3) data files used to update the pbs.org site through the partial update process.

* Recordupdate.xml: Contains the data which will update an existing record, selectively adding and removing dimension and property values.
* ecordadd.xml: Contains the data which will add an entirely new record with a set of property values to an existing index.
* delete.txt: Contains the data which will remove a specific record from an existing index.

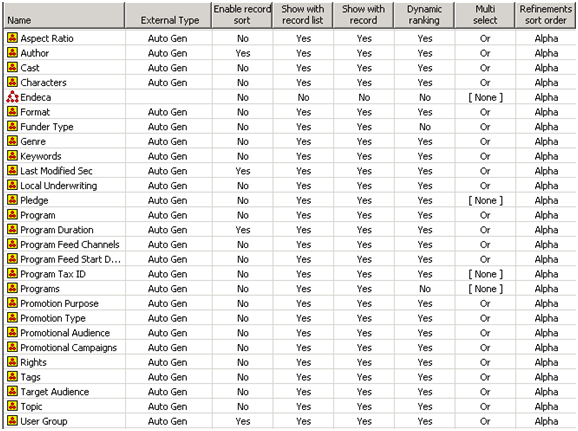
**Data Pre-processing**All data pre-processing is performed before the data is loaded into the E:/Endeca/apps/pbsconnect/data/ektron/full/ directory. Before the data is fed into the pipeline, Thanx has implemented several python scripts to clean and check the data. E:/Endeca/apps/pbs/

|  |  |
| --- | --- |
| Custom Scripts | Operation |
|  |  |
|  |  |

EXAMPLE: Thanx has implemented a custom thesaurus script createThesaurus.bat which converts the thesaurus entries to the Endeca required format and feeds it directly into the IFCR. **Endeca IAP Data Structures  
  
Properties**The screenshot defines the properties to be created for this implementation.



**Dimensions**The table below defines the dimensions to be created for this implementation. The dimensions are dynamically generated from the PBS database and the Attribute dimensions list will change over time. Currently there are 27 dimensions defined and these match the list of attribute values defined in this section. Dimensions configuration is defined within the api\_input configuration files, defined below.



**Dimension Groups**No dimension groups have been designated for configuration at this time. Here is where the list of the groups and the dimensions can be defined:

|  |  |
| --- | --- |
| Group | Dimensions |
|  |  |