Endeca® Commerce Suite

Getting Started Guide

December 2011

Contents

Copyright and disclaimer

Preface

The Endeca® Commerce Suite is the foundation for building applications that help people understand complex information, fostering discovery and improving daily decision-making. These applications instantly summarize data and content for users -- even for unanticipated requests. The Guided Summarization™ experience leads to unexpected insights in millions of everyday decisions, increasing revenue, decreasing costs, and accelerating operations.

The Endeca Commerce Suite is powered by MDEX Engine™ technology, a new class of database designed for exploring information, not managing transactions. The MDEX Engine is supported by:

- The Information Transformation Layer that unites and enriches disparate sources of information while maintaining, augmenting, and even creating structures across the data and content.
- An adaptive application component library that enables the rapid development of information access applications that automatically adapt to changes in the data and content.
- A Web-based management suite that empowers managers to highlight the right information at the right time to end users through adaptive presentation rules and dynamic pages.

These essential capabilities are delivered as an enterprise-class platform, with the scalability, reliability, and security that leading organizations demand.

About this guide

This guide describes the process of installing the core packages of the Endeca Commerce Suite, and provisioning and configuring components.

Important: If you have purchased the Endeca Information Access Platform, please refer to the *Endeca IAP Getting Started Guide*.

This guide walks you through the process of setting up your Endeca implementation based on the Content Assembler reference application. It assumes that you have a basic understanding of the Endeca Commerce Suite products and are familiar with basic Endeca concepts. For more information, see the *Endeca Concepts Guide* and the *Endeca Glossary*, available for download from the Endeca Development Network (EDeN).

Use the *Endeca Commerce Suite Getting Started Guide* to get started with an Endeca Commerce Suite project:

1. Read about the core Endeca Commerce Suite packages and the Endeca Deployment Template, and learn how to download and install them.
2. Run the Deployment Template scripts to provision and initialize a sample application on a single development server.
3. Use the Content Assembler reference implementation to explore basic Page Builder and Content Assembler features with the sample wine application.

This guide also contains information about additional Endeca packages, the documentation for each package, and default Endeca variables and ports.

For detailed installation instructions for each component, see the individual installation guides, available for download from EDeN.

Note: This guide is not a replacement for the *Endeca Content Assembler API Developer's Guide* or the *Endeca Page Builder Template Development Guide*.

Who should use this guide

It is written for application developers who are responsible for building Endeca applications using the Endeca Commerce Suite.

This guide is intended to help Endeca customers through the process of installing the Endeca Commerce Suite. It also provides a walk-through on how to setup and run the reference application.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ¬

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Endeca Customer Support

The Endeca Support Center provides registered users with important information regarding Endeca software, implementation questions, product and solution help, training and professional services consultation as well as overall news and updates from Endeca.

You can contact Endeca Standard Customer Support through the Support section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*.

Part 1

Introduction

Chapter 1

Introduction

This section provides an overview of Endeca and its components.

Welcome to Endeca

The Endeca Information Access Platform consists of three core packages and several optional components. This guide focuses on working with the core packages and the Deployment Template.

The Endeca IAP is comprised of the following core packages:

- Endeca MDEX Engine
- Endeca Workbench
- Endeca Platform Services

Endeca includes many additional components, but this guide is an introduction to setting up the three core packages and using the Deployment Template to manage them.

Core installation packages

The Endeca Commerce Suite consists of several core packages and several optional components. This guide focuses on working with the core packages and the Deployment Template.

Core packages comprising the Endeca Commerce Suite:

- MDEX Engine
- Platform Services
- Merchandising Workbench
- Content Assembler API for Java or Content Assembler API for the RAD Toolkit for ASP.NET

Optional packages:

- Endeca Deployment Template

  Note: This guide assumes that you have downloaded and installed the Endeca Deployment Template.

- Developer Studio
- URL Optimization API for Java
- URL Optimization API for the Endeca RAD Toolkit for ASP.NET

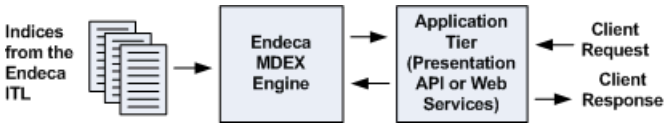
- Endeca Content Acquisition System
- Endeca Document Conversion Module

To determine the compatibility of Endeca installation packages, see the *Endeca InFront Compatibility Matrix* available on EDeN.

MDEX Engine overview

The Endeca MDEX Engine is the indexing and query engine that provides the backbone for all Endeca solutions.

The MDEX Engine uses proprietary data structures and algorithms that allow it to provide real-time responses to client requests. The MDEX Engine stores the indices that were created by the Endeca Information Transformation Layer (ITL). After the indices are stored, the MDEX Engine receives client requests via the application tier, queries the indices, and then returns the results.



The MDEX Engine is designed to be stateless. This design requires that a complete query be sent to the MDEX Engine for each request. The stateless design of the MDEX Engine facilitates the addition of MDEX Engine servers for load balancing and redundancy. Because the MDEX Engine is stateless, any replica of an MDEX Engine on one server can reply to queries independently of a replica on other MDEX Engine servers.

Consequently, adding replicas of MDEX Engines on additional servers provides redundancy and improved query response time. That is, if any one particular server goes down, a replica of an MDEX Engine provides redundancy by allowing other servers in the implementation to continue to reply to queries. In addition, total response time is improved by using load balancers to distribute queries to a replica MDEX Engine on any of the additional servers.

The MDEX Engine package contains the following components:

MDEX Engine Component	Description
Dgraph	The Dgraph is the name of the process for the MDEX Engine. A typical Endeca implementation includes one or more Dgraphs. Optionally, it can include an Agraph that manages a number of Dgraphs.
Agraph	The Agraph is the name of the program that runs in a distributed configuration in addition to the Dgraph. The Agraph typically resides on a separate machine. The Agraph program is responsible for receiving requests from clients, forwarding the requests to the distributed Dgraphs, and coordinating the results. From the perspective of the Endeca Presentation API, the Agraph program behaves similarly to the Dgraph program. Agraph-based implementations allow parallelization of query processing. The implementation of this parallelization results from partitioning the set of records into two or more disjoint

MDEX Engine Component	Description
	subsets of records and then assigning each subset to its own Dgraph. **Note:** Starting with the MDEX Engine version 6.0, (namely, with installations on the 64-bit platforms) a more powerful Dgraph can accommodate much larger data sets without the need to implement an Agraph.
Dgidx	Dgidx is the indexing program that reads the tagged Endeca records that were prepared by Forge and creates the proprietary indices for the Endeca MDEX Engine.
Agidx	Agidx is the program that creates a set of Agidx indices which support the Agraph program in a distributed environment.
dgwordlist	The `dgwordlist` utility is used to manually compile the text-based `worddat` dictionary into the binary `spelldat` dictionary. This enables use of the Aspell dictionary module in the MDEX Engine.
enecerts	The Endeca `enecerts` utility creates the SSL certificates.

Platform Services overview

The Endeca Platform Services package consists of a number of components that are used to build Endeca applications in support of the Endeca MDEX Engine.

Two of the major components of the Endeca Platform Services package are the Endeca Information Transformation Layer (which includes Forge and other Data Foundry components) and the Endeca Application Controller (EAC). The following table lists the components that are available in the Platform Services installation package.

Platform Services Component	Description
Endeca Application Controller (EAC)	The EAC components consist of the EAC Central Server (which coordinates the command, control, and monitoring of all Agents in an Endeca implementation), the EAC Agent (which controls the work of an Endeca implementation on a single host machine) and the EAC command-line utility, `eaccmd`.
Data Foundry	Consists of the Forge program and its related components, such as record adapters, record manipulators, dimension servers, property mappers, and so on. The Content Adapter Development Kit (CADK) is also installed. Note that the Dgidx program is not part of this package, but is available in the MDEX Engine installation package.
Presentation and Logging APIs	APIs to the Endeca MDEX Engine and Log Server. The Endeca Presentation API must be installed on the machine that hosts the Web application server.

Platform Services Component	Description
Logging and Reporting System	The Log Server and Report Generator, which (together with the Logging API) make up the Endeca Logging and Reporting System.
Reference Implementations	Sample Endeca applications that include a sample Developer Studio project (including source data and instance configuration files), as well as JSP and .NET user interface (front-end) applications.
emgr_update	A utility that lets you upload the instance configuration to Endeca Workbench and download it from Endeca Workbench.
Endeca Control System	The Endeca Job Control Daemon (JCD) and the Control Interpreter. These components control and administer the Endeca Information Access Platform running on one or multiple host machines. The Endeca Control System should be installed on the machine that hosts the Endeca Platform Services. Note that the Control System is deprecated, and is not installed by default.

Endeca Workbench overview

Endeca Workbench is a suite of tools that brings together best-in-class Web-site management capabilities including merchandising, Content Spotlighting, search configuration, and usage reporting.

In addition to these powerful tools for business users, Endeca Workbench provides features for system administrators to configure the resources used by an Endeca implementation, monitor its status, start and stop system processes, and download an implementation's instance configuration for debugging and troubleshooting purposes.

The Endeca Workbench package contains the following components:

- Endeca Tools Service
- The appropriate edition of Endeca Workbench for the product you purchased.

In addition, the installation includes a version of the Endeca JSP reference application, which serves as the default preview application in Endeca Workbench.

About the Endeca Page Builder

The Endeca Page Builder is a powerful template-based tool that enables the rapid creation of rich, data-driven pages. The Page Builder is part of the Endeca Workbench suite of tools.

With the Page Builder, content administrators can create landing pages for a particular search term or navigation state based on a set of custom templates. The ability to organize and combine content in a modular fashion within a template framework allows a wide degree of flexibility in crafting pages while maintaining a consistent look-and-feel across an entire site.

For the content administrator, the tool offers a holistic interface to manage the placement and display of content (including rich media, refinements, promotions, Content Spotlighting, and more) all within the overall context of a page, rather than as isolated content sections. Each dynamic page can be used in any number of locations across a site while presenting content that is contextually relevant to an end user's navigation state. This functionality greatly decreases the time and effort normally associated with the creation and maintenance of custom pages.

About the Content Assembler API

The Content Assembler API provides a simple interface to access template-driven content for rendering in Web-based applications.

The Content Assembler API enables a Web application to query the MDEX Engine and retrieve the appropriate content based on a user's navigation state or other triggers. The Content Assembler returns both Endeca query results familiar from the Presentation API or RAD API as well as a content item object that encapulates the page configuration specified by the content administrator. All the content for a page, including the results of any additional queries needed for spotlighting or merchandising, are wrapped in the content item object, simplifying the logic in the front-end application by reducing the need to manage sub-queries in the application layer.

About the Content Assembler reference applications

The Content Assembler package contains a sample application including a sample data set, Endeca project configuration, and front-end application.

The sample project enables you to get a basic implementation up and running quickly, and to explore the functionality that the Content Assembler offers for dynamically composing features into highly flexible, configurable Web applications.

The reference front-end application demonstrates best practices for using the Content Assembler API to develop dynamic applications, and can be used as a starting point for your own application development. It includes a set of cartridges that enable content administrators to configure a variety of application features in the Page Builder, such as Guided Navigation, Content Spotlighting, record results, and other content on a per-page basis.

The Content Assembler reference application is available for Java and for the Endeca RAD Toolkit for ASP.NET.

About the Deployment Template

The Deployment Template provides a collection of operational components that serve as a starting point for development and application deployment.

The template includes the complete directory structure required for deployment, including Endeca Application Controller (EAC) scripts, configuration files, and batch files or shell scripts that wrap common script functionality.

The Deployment Template is the recommended method for building your application deployment environment.

Chapter 2

Installing the Endeca Commerce Suite

This section provides prerequisite and instructional information about installing the Endeca Commerce Suite.

Downloading the Endeca software

You can download the Endeca core software packages from the Downloads section of the Endeca Developer Network (EDeN).

Before downloading and installing the Endeca software:

- Decide on and provide the hardware that will serve as your development environment.
- Decide on which packages of the Endeca software you need to install. In many cases, an Endeca Technical Consultant details for you all the Endeca software packages based on the requirements for your Endeca implementation.
- Consider establishing remote access and a dedicated FTP account on the selected development servers.
- Configure access to source control to secure all Endeca application code and configuration.

To download the Endeca software:

1. Establish a Support account with download access through the Support section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*.
   This enables the Endeca Support and Customer Care groups to track which versions of the software you are using.
2. Download the Endeca core packages on one or more development servers.
3. Download the Endeca Deployment Template from the **DOWNLOADS** > **TOOLS & UTILITIES** section of the Endeca Developer Network (EDeN).

Now you can proceed to install the Endeca packages and set the Endeca environment variables.

Installer file names

Endeca installation packages and executables are named according to a common convention.

The installer file names follow the format:

```
componentname_version_arch-OS
```

For example:

```
mdex_614_x86_64pc-linux.sh
```

The *componentname* is the component identifier for the component being installed. In the example installer, `mdex` is the identifier for Endeca MDEX Engine.

The *version* is the release version, without periods. In the example installer, `614` identifies Endeca MDEX Engine version 6.1.4.

The *arch-OS* is the architecture and operating system identifier for the component being installed. In the example installer, `x86_64pc-linux` identifies the file as an installer for the 64-bit Linux platform. The following table lists the *arch-OS* identifiers and their platforms:

arch-OS identifier	Installation platform
x86_64pc-linux	Linux running on 64-bit Intel processors
sparc_64-solaris	Solaris running on 64-bit Sparc processors
x86_64pc-win32	Windows running on 64-bit Intel processors
i86pc-win32	Windows running on 32-bit Intel processors  **Note:** The Endeca MDEX Engine and Endeca Workbench are only supported on 64-bit based hardware and operating systems. There are no Windows 32-bit installers for the MDEX Engine or Endeca Workbench.

Preparing for installation

Before you install or upgrade any Endeca components, make sure to read installation and migration requirements as they apply to your scenario.

The following documentation can be accessed from the Knowledge Base section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*.

Component	Related Documentation
Endeca MDEX Engine	See the *Endeca MDEX Engine Installation Guide* and the *Endeca MDEX Engine Migration Guide*.
Endeca Presentation API	See the *Installation instructions and release notes* file.
Endeca Workbench	See the *Endeca Workbench Installation Guide* and the *Endeca Workbench Migration Guide*.
Endeca Platform Services	See the *Endeca Platform Services Installation Guide* and the *Endeca Platform Services Migration Guide*.

Component	Related Documentation
Endeca Content Acquisition System	See the *Endeca Content Acquisition System Installation Guide* and the *Endeca Content Acquisition System Migration Guide*.

Installing the Endeca Commerce Suite components

This section provides high-level installation instructions for the Endeca Commerce Suite.

Important: If you are upgrading from previous releases of the Endeca software, read the appropriate *Migration Guide* and follow guidance on how to prepare your implementation for migration. Next, proceed with downloading and installing the software packages in the order listed in this topic.

This guide assumes that you are going to install on a single development server. In a typical production environment, it is recommend that you distribute the Endeca components across several servers. While you can install the Endeca packages in any order, Endeca recommends that you install them in the following order:

To install the Endeca Commerce Suite:

1. Install the MDEX Engine package.
2. Install the Platform Services package.
3. Install the Merchandising Workbench package.
4. Install the Content Assembler API.
5. Install the Endeca Deployment Template.
6. Set up a directory for your deployment, such as `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.
7. Install additional packages, such as Developer Studio and Content Acquisition System (CAS) on those servers that require them.
8. Install any separately licensed packages, such as the URL Optimization API, Relationship Discovery, Analytics, CMS connectors, or Document Conversion Module.

Note: For installation instructions, see the appropriate installation guide.

Downloading and installing the Endeca Deployment Template

You can download and run the Deployment Template on a single machine that serves as your Endeca development server, or on several machines running on the same operating system (Windows or UNIX).

Before downloading the Deployment Template, decide on and provision the hardware that you will use in your development environment. To download the template, establish a support account with the Endeca Customer Support Center on EDeN.

For detailed information about the Endeca Deployment Template, see the *Deployment Template Usage Guide*.

To download and install the Endeca Deployment Template:

1. Download the latest available version of the Deployment Template from the Downloads section of EDeN.
2. Unzip the Deployment Template into `C:\` if you are running on Windows or a directory such as `/localdisk/` on UNIX.
   The package creates a directory structure under `C:\Endeca\Solutions` on Windows and `/localdisk/Endeca/Solutions` on UNIX.
3. Create a directory for deploying your project, for example, create `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.

You have installed the Deployment Template and are prepared to run it.

Next, you run the Deployment Template to establish the Endeca project based on the sample wine application, and run a baseline update script in this project.

Setting the Endeca environment variables

Having the environment variables properly set ensures that the different Endeca components can communicate with each other.

To set the Endeca environment variables:

Do the following:

Option	Description
On Windows	To set the environment variables for Platform Services and Endeca Workbench, run the installation process for these packages. This properly sets up the environment variables for them. To set the environment variables for the MDEX Engine (in particular, to set the `ENDECA_MDEX_ROOT`), run the `\Endeca\MDEX\<version>\mdex_setup.bat` script.
On UNIX	Depending on the package and your platform, use the `source` command to run the scripts that set the variables. For example, in your Endeca installation directories, run: • `source /endeca/MDEX/<version>/mdex_setup_sh.ini`. The `mdex_setup` script sets up the MDEX Engine variables. • `source /endeca/PlatformServices/workspace/setup/installer_sh.ini`. This script sets up the Platform Services variables. • `source /endeca/Workbench/workspace/setup/installer_sh.ini`. This script sets up the Endeca Workbench variables.

For information on setting environment variables required by other Endeca packages, refer to the installation guides for each package.

Starting the Endeca HTTP and Tools services

If you have multiple servers, the Endeca HTTP service must be running on all the machines in your Endeca environment, except the Application server. When the Endeca HTTP service is running this means that the Endeca Application Controller (EAC) is running. The Endeca Tools Service must be running on the Tools server.

Before starting the Endeca HTTP and Tools services, verify that you have:

- Installed the MDEX Engine, Platform Services and Endeca Workbench.
- Set the environment variables for the MDEX Engine and Platform Services.

To start the Endeca HTTP service and the Endeca Tools Service:

Do the following:

Option	Description
On Windows	Go to **Start** > **Control Panel** > **Administrative Tools** > **Services**, select the Endeca HTTP service and the Endeca Tools Service and click **Start**.  **Note:** On the servers on which you have installed the Platform Services and Endeca Workbench packages, the Endeca HTTP and Tools services are started automatically when you reboot the machines.
On UNIX	To start the Endeca HTTP service, run `$ENDECA_ROOT/tools/server/bin/startup.sh` To start the Endeca Tools Service, run `$ENDECA_TOOLS_ROOT/server/bin/startup.sh`

Part 2

Working with the sample application

Chapter 3

Running the Sample Application

To configure the sample application, install all the required Endeca packages and the Endeca Deployment Template, and run the Deployment Template scripts to create, provision, and initialize the Endeca application and run the baseline update.

Configuring the sample application

To configure the application, run the Deployment Template `deploy` script.

Before running the Deployment Template, verify that:

- You have installed the MDEX Engine, Platform Services (including the EAC Central Server and Agent), Endeca Workbench, and the Content Assembler API on the same machine.
- The Endeca HTTP and Tools services are running on this server. (When the Endeca HTTP service is running, the EAC is running.)
- You have downloaded the Deployment Template on this server, and set up a directory for your deployment, such as `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.

To configure the application:

1. Open a command prompt or UNIX shell and navigate to the `bin` directory of your Deployment Template installation directory.
   For example:
   - (Windows) `C:\Endeca\Solutions\deploymentTemplate-`*`version`*`\bin`
   - (UNIX) `/usr/local/Endeca/Solutions/deploymentTemplate-`*`version`*`/bin`
2. Run the Deployment Template's `deploy.bat` or `deploy.sh` script, specifying the full path to the sample application's deployment configuration file.
   For example:
   - (Windows with the Content Assembler API for the RAD Toolkit for ASP.NET installed)
     ```
     deploy.bat --app C:\Endeca\ContentAssemblerAPIs\RAD Toolkit for
     ASP.NET\version\reference\pagebuilder_wine_app\deploy.xml
     ```
   - (UNIX with the Content Assembler API for Java installed)
     ```
     deploy.sh --app /usr/local/Endeca/ContentAssemblerAPIs/Java/version/ref¬
     erence/pagebuilder_wine_app/deploy.xml
     ```

Note: If you specify the full path to the sample application's `deploy.xml` file incorrectly, the Deployment Template runs the default script with an error message similar to the following:

```
02/05/2009 16:35:50 [deploy.pl] ERROR: Invalid application descriptor
file specified: [incorrectly specified path]
```

3. Confirm the correct version of the Platform Services installation package (the template verifies the `ENDECA_ROOT` variable), and press `Enter` to proceed.
4. Enter `Yes` to install the base application.
5. Specify a name for the application, for example: `ContentAssemblerRefApp`.
6. Specify the location of the application directory: `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.

Note: In this guide, the directory for each of your applications is referred to by the *`[appDir]`* abbreviation. With the paths above, this is equal to `C:\Endeca\apps\ContentAssemblerRefApp` on Windows and `/localdisk/apps/ContentAssemblerRefApp` on UNIX.

7. Specify the EAC port (the Endeca HTTP service port). To use the default port, specify `8888`.
8. Specify `Y` to enable Workbench integration.
9. Specify the Workbench port (this is the Endeca Tools Service port for your Endeca Workbench edition). To use the default port, specify `8006`.
10. Specify the host name of the Content Assembler reference application or accept the default: `localhost`.
11. Specify the server port for the Content Assembler reference application (for .NET, the default is `80`; for Java, the default is the same as the Workbench port).
12. Specify other necessary ports:
    a) For the Dgraph1, specify the Dgraph1 user query port or accept the default: `15000`.

Note: The Content Assembler reference UI implementation assumes that you are using the default Dgraph port of 15000. If you specify a different port, you need to configure the reference application to connect to the new port. For details, see the *Content Assembler API Developer's Guide*.

    b) For the Dgraph2, specify the Dgraph2 user query port or accept the default: `15001`.

    By default, the Content Assembler `deploy.xml` provisions a project with only one Dgraph. If you prefer to configure two Dgraphs, edit the *`[appDir]`*`/config/script/AppConfig.xml` file to uncomment the `Dgraph2` entries.

    c) For the Endeca Logging and Reporting Server, specify the server port or accept the default: `15010`.

    The script provisions a Logging and Reporting Server; however, the reference UI implementation does not include any calls to the Logging API. For more information about implementing reporting, see the *Log Server and Report Generator Guide*.

Now you have provisioned the directories for the application and need to initialize it.

Initializing the application

To initialize the application, run the `initialize_services` script from the Endeca Deployment Template.

It is assumed that you have run the Deployment Template `deploy` script to create the directory structure, configuration files and scripts for the application.

To initialize the application:

On the development server, or on a Data Processing (ITL) server in your environment, run *[appDir]*`\control\initialize_services.bat` on Windows or *[appDir]*`/control/initialize_services.sh` on UNIX.
This script initializes the sample wine application.

After you have provisioned and initialized the application, you can run the baseline update script using the Deployment Template and also access Endeca Workbench to check the status of the running components.

Running the baseline update script

The baseline update script runs the MDEX Engine (the indexer and the Dgraph) to index the records and to update the MDEX Engine with the indexed data.

Before running the baseline update script, ensure that you have provisioned the sample wine reference implementation with the Deployment Template, by running its *[appDir]*`\control\initialize_services.bat` or *[appDir]*`/control/initialize_services.sh` script.

To run a baseline update script on the Data Processing (ITL) server:

1. Run *[appDir]*`\control\load_baseline_test_data.bat` or *[appDir]*`/control/load_baseline_test_data.sh`
   This script uploads the reference implementation data into the locations expected by the Deployment Template workflow, and communicates to the EAC that the data is ready for processing.
2. Run *[appDir]*`\control\baseline_update.bat` or *[appDir]*`/control/baseline_update.sh` script.
   This script takes a few moments to complete.

In addition to running the baseline update script, you can use the Deployment Template to run a partial update script, a configuration update script , and Log Server scripts for obtaining daily log reports. For information on using and customizing the Deployment Template, see the *Deployment Template Usage Guide*.

Uploading templates to the Page Builder

The sample application provides a set of example templates; you must upload these to Page Builder using the emgr_update utility.

To upload the sample templates:

1. Open a command prompt or UNIX shell.

2. Run emgr_update with the `--action` of `set_templates` and the following parameters:

Parameter	Value
`--host`	The machine name and port for the staging Endeca Workbench environment, in the format *host*:*port*.
`--app_name`	The name of the application to which you want the templates to apply.
`--dir`	The path to the local directory where your templates are stored. The sample templates are located in the `\config\page_builder_templates` directory of your application.

The following is a Windows example:

```
emgr_update.bat --action set_templates --host localhost:8006
--app_name ContentAssemblerRefApp --dir c:\Endeca\apps\ContentAssembler¬
RefApp\config\page_builder_templates
```

The following is a UNIX example:

```
emgr_update --action set_templates --host localhost:8006
--app_name ContentAssemblerRefApp --dir /localdisk/apps/ContentAssembler¬
RefApp/config/page_builder_templates
```

3. (Optional) Confirm that the templates are available in the Page Builder.
   a) Log in to Merchandising Workbench.

      Recall that the default URL of Merchandising Workbench is *http://localhost:8006* (replace *localhost* with the host name of of the server that is hosting Workbench). The default **Username** is `admin` and the default **Password** is `admin`.

   b) Select **Page Builder** from the menu.
      The **List View** displays.

      Note: If you accessed Merchandising Workbench through `localhost`, you may see a warning about JavaScript domain settings. This message can be safely ignored as long as you do not need to use the preview application. You can avoid this message by specifying the same host name in the URL when you log in to Workbench as the host name that is specified for the preview application in the **Application Settings** section.

   c) Click **New Landing Page**.
      The **Edit View** displays.
   d) Click **Select Template**.

   If the **ThreeColumnNavigationPage** template is available, your template upload was successful and you can log out of Merchandising Workbench without saving changes.

If templates do not display in the Page Builder after uploading them using emgr_update, check the Workbench log for possible causes:

- Windows: `%ENDECA_TOOLS_CONF%\logs\webstudio.log`
- UNIX: `$ENDECA_TOOLS_CONF/logs/webstudio.log`

Configuring template thumbnail images in Page Builder

Thumbnail images must be hosted on a Web server in order to display. While thumbnail images may be hosted on a separate Web server from your Page Builder instance, this section assumes that you are hosting the images on the same server as your Page Builder instance.

You must edit the `thumbnails.xml` file to contain the absolute path to the thumbnail images included with the Content Assembler reference application.

To configure thumbnails:

1. Stop the Endeca Tools Service.
2. Navigate to the directory where the `thumbnails.xml` file is located in your Content Assembler API installation, for example:
   - Java: `C:\Endeca\ContentAssemblerAPIs\Java\`*`version`*`\reference`
   - .NET: `C:\Endeca\ContentAssemblerAPIs\RAD Toolkit for ASP.NET\`*`version`*`\reference\context_files`
3. Edit the `thumbnails.xml` file so that the `docBase` points to the `reference\thumbnails` subdirectory of your Content Assembler API directory, for example:
   - Java: `C:\Endeca\ContentAssemblerAPIs\Java\`*`version`*`\reference\thumbnails`
   - .NET: `C:\Endeca\ContentAssemblerAPIs\RAD Toolkit for ASP.NET\`*`version`*`\reference\thumbnails`
4. Copy the `thumbnails.xml` file to `%ENDECA_TOOLS_CONF%\conf\Standalone\localhost` (Windows) or `$ENDECA_TOOLS_CONF/conf/Standalone/localhost` (UNIX).
5. Start the Endeca Tools Service.

Chapter 4

Setting Up the Reference Application

This section describes the process for deploying the Content Assembler reference application.

Deploying the Content Assembler reference application for Java

Configure the Endeca Tools Service for the Content Assembler reference application using the sample context file included in the Content Assembler API installation.

To deploy the Content Assembler reference application for Java:

1. Stop the Endeca Tools Service.
2. Navigate to the `reference` subdirectory of your Content Assembler API directory.
   For example: `C:\Endeca\ContentAssemblerAPIs\Java\`*`version`*`\reference`.
3. Edit the `ContentAssemblerRefApp.xml` file so that the `docBase` points to the `ContentAssemblerRefApp` subdirectory of your Content Assembler API directory.
   For example,
   `C:\Endeca\ContentAssemblerAPIs\Java\`*`version`*`\reference\ContentAssemblerRefApp`
4. Copy the `ContentAssemblerRefApp.xml` file to `%ENDECA_TOOLS_CONF%\conf\Standalone\localhost` (Windows) or `$ENDECA_TOOLS_CONF/conf/Standalone/localhost` (UNIX) directory.
5. Start the Endeca Tools Service.
6. (Optional) Open a Web browser and navigate to your Content Assembler reference application to verify the deployment: *http://localhost:8006/ContentAssemblerRefApp*. Replace *localhost* with the host name or IP address of the server running the reference application. Replace *8006* with the Endeca Tools Service port if it is not running on the default port.

Deploying the Content Assembler reference application for RAD.NET

These instructions assume a typical system configuration that includes IIS 5.1, IIS 6 Manager, and the .NET Framework 2.0. There may be minor configuration differences if you are using other versions of IIS, IIS Manager, or the .NET Framework.

Note: Make sure that you have enabled the ASP.NET 2.0 Web Service Extension in IIS before deploying the reference application.

To deploy the Content Assembler reference application for the RAD Toolkit for ASP.NET:

1. From the Windows Control Panel, select **Administrative Tools** > **Internet Information Services (IIS6) Manager**.
2. In the IIS tree pane, expand the machine icon for the local machine, then expand the **Web Sites** directory.
3. Right click the **Default Web Site** and select **New** > **Virtual Directory...**.

   Note: If you are using IIS 7, you should create an **Application** rather than a **Virtual Directory**.

4. Complete the **Virtual Directory Creation Wizard** as follows:
   a) Click **Next**.
   b) Type an alias name such as `ContentAssemblerRefApp`.

      Note: If you intend to use the image preview feature for the reference application in the Page Builder, the sample templates assume that the application is deployed in a directory named `ContentAssemblerRefApp`.

   c) Click **Next**.
   d) In the **Web Site Content Directory** screen, click **Browse** and locate the reference application that is packaged with the Content Assembler API for the RAD Toolkit for ASP.NET. In a typical installation, this is in `C:\Endeca\ContentAssemblerAPIs\RAD Toolkit for ASP.NET\`*`version`*`\reference\ContentAssemblerRefApp`.
   e) Click **Next**.
   f) In the **Access Permissions** window, leave the default settings in place.
   g) Click **Next**, and then click **Finish**.
5. In the IIS tree pane, expand the machine icon and locate the virtual directory named `ContentAssemblerRefApp` that you created in the step above.
6. Right click `ContentAssemblerRefApp` and select **Properties**.
7. Select the **Virtual Directory** tab and perform the following tasks:
   a) Under the **Application Settings** section, click **Create**.
   b) From the **Execute permissions** list, select **Scripts only**.
   c) Click **Apply**.
8. Select the **Documents** tab and perform the following tasks:
   a) Check **Enable default content page**.
   b) Click **Add...**.
   c) In the **Default content page** field, type `Content.aspx`.
   d) Click **OK**.
   e) Select `Content.aspx` and click **Move Up** until the file is at the top position.
9. Select the **ASP.NET** tab and from the **ASP.NET version** list, select 2.0.x or later.
10. Restart IIS.
    a) With your Web site selected in the tree pane, click **Stop Item** in the toolbar, then click **Start Item**.

11. (Optional) Open a Web browser and navigate to your Content Assembler reference application to verify the deployment: *http://localhost/ContentAssemblerRefApp/Content.aspx*.

    Replace *ContentAssemblerRefApp* with the name of the virtual directory in IIS.

Making the Content Assembler reference application your preview application

The preview application in the Page Builder allows content administrators to verify the behavior of the pages they create.

Before content administrators create pages in the Page Builder, you should replace the default preview application with one that can render pages appropriately based on the templates you have created. You should update the preview application periodically as you begin to customize cartridges.

To make the Content Assembler reference application your Merchandising Workbench preview application:

1. Log in to Merchandising Workbench as an administrative user.
2. Navigate to **Application Settings** > **Preview App settings**.
3. If your reference application is running on a different host or port than Merchandising Workbench, set the **Javascript domain** for the Merchandising Workbench.
4. Edit the default **URL Mappings** as indicated below to point to your Content Assembler reference application. The Page Builder does not make use of the **Record URL**, so you can leave the default value.

   In the examples below, replace *host* with the fully qualified domain name of the machine running the reference application. If you are not running the application in the default context, replace the `/ContentAssemblerRefApp/` in the path.

   For the Content Assembler reference application for Java, specify the following:

Search URL	`http://host:8006/ContentAssemblerRefApp/controller?N=0&Ntk=${key}&Ntt=${terms}&Nty=1&D=${terms}&Ntx=mode+matchallpartial&Dx=mode+matchall&Nmpt=${previewtime}&Nmrf=${rulefilter}`
Navigation URL	`http://host:8006/ContentAssemblerRefApp/controller?N=${nav}&Nmpt=${previewtime}&Nmrf=${rulefilter}`
Search and Navigation URL	`http://host:8006/ContentAssemblerRefApp/controller?N=${nav}&Ntk=${key}&Ntt=${terms}&Nty=1&D=${terms}&Ntx=mode+matchallpartial&Dx=mode+matchall&Nmpt=${previewtime}&Nmrf=${rulefilter}`

   For the Content Assembler reference application for the RAD Toolkit for ASP.NET specify the following:

Search URL	`http://host:80/ContentAssemblerRefApp/Content.as¬ px?dsNav=Ntk:${key}%7c${terms}%7c1%7c,Nmpt:${previewtime},Nm¬ rf:${rulefilter}`
Navigation URL	`http://host:80/ContentAssemblerRefApp/Content.as¬ px?dsNav=N:${nav},Nmpt:${previewtime},Nmrf:${rulefilter}`
Search and Navigation URL	`http://host:80/ContentAssemblerRefApp/Content.as¬ px?dsNav=Ntk:${key}%7c${terms}%7c1%7c,N:${nav},Nmpt:${preview¬ time},Nmrf:${rulefilter}`

Note: Be sure to include the port number even if you are using the default HTTP port of 80.

Specifying the fully qualified domain name in the URL ensures that the preview application is enabled for users logging in to Merchandising Workbench from any machine. If you happen to be running your browser locally to the Workbench server, you can log in to Merchandising Workbench with a URL like *http://localhost:8006*. However, if you use localhost to connect, a Javascript warning displays in the Page Builder. To avoid the warning, connect to Merchandising Workbench using the fully qualified domain name.

5. Click **OK**.
6. If your reference application is running on a different host or port than Merchandising Workbench, you must set the Javascript domain in the reference application.
   - For the Content Assembler API for the RAD Toolkit for ASP.NET reference application:
     1. Navigate to the `reference\ContentAssemblerRefApp` subdirectory of your Content Assembler API installation directory.
     2. Open the `Content.aspx` file and locate the following line:
        ```
        // document.domain = "localhost:3050";
        ```
     3. Uncomment the line and edit the value of `document.domain` to match the Javascript domain you set in Merchandising Workbench.
     4. Restart IIS.
   - For the Content Assembler API for Java reference application:
     1. Stop the Endeca Tools Service.
     2. Navigate to the `reference\ContentAssemblerRefApp\layout` subdirectory of your Content Assembler API installation directory.
     3. Open the `ThreeColumnNavigationPage.jsp` file and locate the following line:
        ```
        // document.domain = "localhost:8006";
        ```
     4. Uncomment the line and edit the value of `document.domain` to match the Javascript domain you set in Merchandising Workbench.
     5. Start the Endeca Tools Service.

For more details about preview application settings, please refer to the *Endeca Merchandising Workbench Help*.

Javascript domain requirements

If Merchandising Workbench and your preview application do not reside on the same host and port, you must declare the Javascript domain in the preview application's code.

Merchandising Workbench communicates with and controls the preview application via Javascript. As a result, both Merchandising Workbench and the preview application must have the same Javascript domain property. The domain property provides security for scripts that run in different browser windows but need to communicate with one another.

To enable the Page Builder to communicate with the preview application, you must declare the domain in the navigation results page (the page that shows the set of records that correspond to a user's query).

When you specify the Javascript domain, you can also include the port number of the application server. This ensures that you are referring to the exact host machine and port number. For example, if the custom application is on an application server running on port 8080, you can declare the Javascript domain with the following:

```
document.domain = "example.com:8080";
```

You can also use the host machine's IP address, the fully qualified domain name, the machine name (such as `web004`), or `localhost`, but be aware that different browsers have varying support for the different host name formats.

Note: In addition, Merchandising Workbench's **Preview App settings** page provides a field where you must enter this information. This is analogous to declaring the domain in your Javascript headers.

For information about setting up a custom application as the Page Builder preview application, please refer to the *Endeca Workbench Administrator's Guide*.

Seeing the Page Builder in action

Once you have completed setup of the sample application, you can log in to the Page Builder, make edits to a landing page, and see the changes in the reference application.

Before proceeding ensure that you have run a baseline update, uploaded the sample templates to Page Builder, and deployed the Content Assembler reference application.

To validate the sample project:

1. Open a Web browser and navigate to your Content Assembler reference application.
   - For Java: *http://localhost:8006/ContentAssemblerRefApp*. Replace *localhost* with the host name or IP address of the server running the reference application. Replace *8006* with the Endeca Tools Service port if it is not running on the default port.
   - For .NET: *http://localhost/ContentAssemblerRefApp/Content.aspx*. Replace *localhost* with the host name or IP address of the server running the reference application. Replace*ContentAssemblerRefApp* with the name of the virtual directory in IIS.
2. Using the navigation menu on the left, navigate to **Wine Type** > **Red**.
   Notice the site banner image and the promotions in the right column.
3. Log in to the Merchandising Workbench.
   a) Open a second Web browser window and navigate to *http://localhost:8006*

Replace *localhost* with the host name or IP address of the server running Merchandising Workbench. Replace *8006* with the Endeca Tools Service port if it is not running on the default port.

b) Enter the default **Username** `admin` and **Password** `admin`.
c) Click **Log in**.

4. Select **Page Builder** from the menu.
   The **List View** displays.
5. Select **Red Wine Landing Page** from the landing pages list.
   The **Edit View** displays.
6. From the **Content Tree** on the left, expand the **Header** section and choose **Site Banner**.
7. From the **Content Detail Panel** on the right, edit the **Image name** to read `/images/WineDemoImages/site-banner-bottles.jpg`.
8. From the **Content Tree** on the left, expand the **RightColumn** section.
9. Reorder the list by dragging **Customer Favorites** to the bottom of the list and **Wine Club** to the top.
10. Click **OK**.
    You are returned to the **List View**.
11. Click **Save All Changes**.
    The **Status** dialog box displays.
12. Click **OK**.
13. Switch to the Web browser window with your Content Assembler reference application. Refresh this window to view your changes.
    The banner image of wine barrels is replaced by a row of bottles, and the order of items in the right column has changed.
14. Click the **Wine Store** logo to return to the home page.
    Note that the change to the site banner applies only to the red wine page.

Part 3

What's Next

Chapter 5

Guide to Endeca Documentation

This section provides information about locating the documentation required for configuring various aspects of an Endeca implementation.

Where to find relevant documentation

This topic lists Endeca Commerce Suite documentation relevant to each major implementation task.

For information about	See this documentation
Basic Endeca concepts	• *Concepts Guide* • *Glossary*
Data ingest	The CAS documentation set, especially: • *CAS Quick Start Guide* • *CAS Developer's Guide* • *CAS Console for Endeca Workbench Help*
Pipeline creation	• *Forge Guide* • *Developer Studio Help* • *Partial Updates Guide*
The Endeca Workbench interface, including Page Builder	• *Endeca Workbench Help* • *Endeca Workbench User's Guide*
Configuring Endeca Workbench	• *Endeca Workbench Administrator's Guide*
Developing applications with Endeca Page Bulider and the Content Assembler API	• *Page Builder Developer's Guide* • *Content Assembler API for Java Developer's Guide* • *Content Assembler API for the RAD Toolkit for ASP.NET Developer's Guide*

For information about	See this documentation
Developing applications with the Endeca URL Optimization API	• *URL Optimization API for Java Developer's Guide* • *URL Optimization API for the RAD Toolkit for ASP.NET Developer's Guide*
Deployment and operational tasks	• *IAP Administrator's Guide* • *Deployment Template Usage Guide*

Accessing documentation on EDeN

All Endeca documentation is available for download or browsing on the Endeca Developer Network (EDeN).

In order to access EDeN, you must have a registered account. If you do not have an account, contact your Endeca administrator.

To access the documentation on EDeN:

1. In a Web browser, navigate to *http://eden.endeca.com*.
2. Log in using your username and password.
3. From the top menu, click **Knowledge Base**.
4. From the drop-down list, select the product and version you are interested in.
5. Click on a heading to expand or collapse a category.
6. Click the title of a document to browse its contents online, or click the **[Download]** link to save a printable version of the document for offline reading.

Chapter 6

Additional Endeca Packages

In addition to the core packages, there are several additional installation packages and a number of separately licensed packages.

Additional installation packages

The following packages are highly recommended, although not required to follow the procedures in this guide. They can be installed and integrated into your implementation later. Access to these packages is included with all licenses of the Endeca IAP.

About Developer Studio

Developer Studio is a Windows application that you use to define all aspects of your instance configuration including pipeline components, Endeca properties and dimensions, precedence rules, dynamic business rules, and user profiles.

With Developer Studio, you can define:

- Pipeline components for tasks such as loading, standardizing, joining, mapping, and exporting data.
- Endeca properties and property attributes such as sort and rollup.
- Dimensions and dimension values, including dimension hierarchy.
- Precedence rules among dimensions that provide better control over your implementation's navigation flow.
- Search configurations, including which properties and dimensions are available for search.
- Dynamic business rules that allow you to promote certain records on your Web site using data-driven business logic. Dynamic business rules are used to implement merchandising and content spotlighting.
- User profiles that tailor the content returned to an end-user based upon pre-configured rules.

Developer Studio uses a project file, with an `.esp` extension, that contains pointers to the XML files that support an instance configuration.

About the Content Acquisition System (CAS)

The Content Acquisition System (CAS) provides components that manage all file system and CMS crawls, as well as all Web crawls.

The CAS package includes:

- Endeca CAS Server
- Endeca CAS Console
- Endeca CAS API
- Endeca Web Crawler.

Note: Connectors to a variety of content management systems (CMSs) are available as separately licensed packages.

About the RAD Toolkit

The Rapid Application Development (RAD) Toolkit provides controls and components to build Endeca applications and also provides a simplified interface to the Endeca Presentation API. The RAD Toolkit is available for ASP.NET.

The RAD Toolkit for ASP.NET contains the following components:

RAD Toolkit for ASP.NET component	Description
RAD API for .NET	Provides a simplified interface to the Endeca MDEX Engine and makes programming more friendly to the typical .NET developer.
Visual Studio server controls, including Endeca data source controls, and Endeca user interface controls	These controls help developers quickly build Endeca applications and also provide a simple interface to the Endeca Presentation API for ASP.NET. The controls participate in ASP.NET declarative data binding and include an Endeca-specific data source control to easily set host, port, and query-specific information.
Reference application	Like other Endeca reference applications, the RAD Toolkit reference application provides a simple front-end interface that allows you to connect to an MDEX Engine and examine a record set. This reference application can be run in Postback mode, URL mode, RAD Toolkit Server Controls mode, or RAD Toolkit Server Controls URL mode.

Separately licensed packages

The following packages are sold separately. For more information about these packages, contact your Endeca representative.

Separately licensed package name	Description
Analytics and Visualization	Extends MDEX Engine capability by enabling mathematical calculations and aggregations on data in an Endeca index. These aggregated results can be seamlessly integrated with the other parts of an Endeca application. This module includes a set of visualization tools that enables the display of interactive graphs, charts, and histograms of data returned in Endeca queries, with integrated support in the Endeca tool set. Endeca Analytics is the next generation of dynamic summarization, building on the power of Guided Navigation. The software for the Analytics and Visualization package is installed as part of the MDEX Engine core package. The functionality is enabled via the MDEX Engine product configuration file.
Advanced Query Module for Record Relationship Navigation and XQuery	Enables Record Relationship Navigation, Custom Web Services, and XQuery. • RRN allows for Guided Navigation on interrelated record types so that a user can have multiple perspectives on the data. The software for Record Relationship Navigation is installed as part of the MDEX Engine core package. • Custom Web Services allow applications and data services to integrate cleanly with SOA Infrastructure and with non-Java and non-.NET development environments. • XQuery enables developers to customize query execution within the MDEX Engine through the development of stored procedures.
Relationship Discovery (Term Discovery and Cluster Discovery)	These components enable the discovery of latent structure in unstructured text data. Term Discovery applies linguistic and statistical techniques to automatically extract significant concepts and noun phrases from text. These important terms can be presented as refinements in the search and navigation experience or can be used as input to the Cluster Discovery component, which automatically groups terms into related subsets to create clusters of related documents or records. This module is available for English and French language text as separately priced options. The software for the Relationship Discovery package is installed as part of the Platform Services core package. The functionality is enabled via the Platform Services product configuration file.
Text Enrichment	Includes Relationship Discovery as well as advanced text analysis capabilities for extraction of people, places, organizations, quotes, and themes. The module also includes core summarization capabilities for automatic creation of abstracts or topical summaries.

Separately licensed package name	Description
	The software for Text Enrichment is available from the support section of EDeN.
Text Enrichment with Sentiment Analysis	Includes Relationship Discovery as well as additional advanced text analysis capabilities for extraction of people, places, organizations, quotes, themes and the aggregate sentiment related to each of these. Sentiment is extracted with a score indicating the positive and negative nature of a document or an entity in the document. These scores can be used for guided navigation by varying ranges of positivity and negativity. This module also includes core summarization capabilities for automatic creation of abstracts or topical summaries. The software for Text Enrichment with Sentiment Analysis is available from the support section of EDeN.
DataLens Foundry	This module uses unique semantic-based technology from Endeca partner Silver Creek Systems to provide an integrated interface with data cleansing and normalization tools to provide higher quality, more highly attributed data for your Endeca implementation.
Document Conversion Module	Extends Platform Services and lets you convert source documents from a variety of file formats to text. Allows indexing of text stored in over 390 different file formats, covering common formats such as Word, Excel, Acrobat, and a host of others. You can manipulate the resulting text as part of Data Foundry processing for use in Endeca records. The software for this package includes conversion libraries that you can add to the Platform Services core package.
Enterprise Application Adapters	These separately priced modules allow indexing of content stored in various enterprise applications. Currently supported systems include: SAP ERP, SAP BW, Siebel, PeopleSoft, and JD Edwards.
CMS Connectors	Extend the CAS Server by providing connectors that enable crawling and indexing of content stored in various content management systems (CMS). Examples of supported systems include: EMC Documentum, EMC Documentum eRoom, FileNet P8, FileNet Document and Image Services, Hummingbird DM, Interwoven TeamSite, JSR-170 compliant, Lotus Notes/Domino, MS SharePoint 2003/2007, and OpenText LiveLink. The software for these packages is installed as part of the CAS core package. The functionality is enabled via the CAS product configuration files.

Separately licensed package name	Description
East Asian Language Support Modules	Improve search recall with stemming and also include a morphological tokenization or segmentation capability for Chinese, Japanese, and Korean. Base software is installed as part of the MDEX Engine core package. Separate license keys are required to use the language functionality. Each Language Support Module is a separate additional-cost option.
Language Support Modules for stemming	Improve keyword search recall by adding a stemming capability for French, Italian, German, Spanish, Portuguese, and Dutch. Base software is installed as part of the MDEX Engine core package. Separate license keys are required to use the language functionality. Each Language Support Module is a separate additional-cost option.
Merchandising Workbench	A management tools suite that enables merchandisers to rapidly create differentiated shopping experiences and extract new value from their Endeca investments through increased sales, higher ROI on Search Engine Marketing, greater customer loyalty, and reduced site management costs. The Merchandising Workbench includes best-in-class capabilities for merchandising, landing page management, search configuration, reporting, and taxonomy editing.
Publishing Workbench	A management tools suite similar to Merchandising Workbench, but targeted to editors of Media and Publishing sites.
Social Navigation	Gives site owners new ways to take advantage of user-generated content, including product reviews, opinions, ratings, and user-generated tags, as well as contributor profiles. Using this data, the IAP helps people discover information and products based on the paths and contributions of individuals with similar needs and preferences. This module includes a limited set of adapters to best-of-breed social media software, template application code, and a limited-use license of the Record Relationship Navigation and English Relationship Discovery capabilities.
Search Engine Optimization	Improves both organic and paid strategies, driving qualified traffic and boosting content consumption, click-through rates and conversion rates. Included in the module is the Sitemap Generator, which makes dynamic pages appear to be static when crawled or indexed by major Internet search engines. Also included is the automated URL Optimization API, which turns dynamic page URLs into SEO-friendly text-based URLs that give the appearance of static, keyword-rich directory pages.

Separately licensed package name	Description
Allurent Rich Interface Module	Flexible and fully integrated user interface components that offer a compelling shopper experience without the need for expensive redesign projects. Consists of two separately priced components: Allurent Navigation for integrated search and browse, and Allurent Details for in-context presentation of product detail information.

Appendix A

Full List of Documentation Resources

This section describes the documentation related to each platform component. Only essential documentation is included with the product installation, but all Endeca documentation is available on the Endeca Developer Network (EDeN) for browsing or download, either individually or as part of an overall Documentation package.

General Endeca documentation

The following table lists the documentation that applies across multiple Endeca packages.

Title	Description
Endeca Getting Started Guide	Overview of Endeca components including information about configuration scenarios.
Endeca Concepts Guide	Introduction to the Endeca Information Access Platform. Covers the key concepts underlying Endeca applications.
Endeca Administrator's Guide	Describes tasks involved in administering and maintaining applications built upon the Endeca Information Access Platform. It bridges the gap between the work performed by the Endeca Services team and the issues that system administrators encounter when maintaining the system.
Endeca Glossary	A reference for Endeca terms and definitions.
Endeca Third-Party Software Usage and Licenses	Provides copyright, license agreement, and/or disclaimer of warranty information for the third-party software packages that Endeca incorporates.

MDEX Engine documentation

The following table lists the documentation related to the MDEX Engine package.

Title	Description
Analytics Guide	Provides an overview of Endeca Analytics and describes the Analytics and Charting APIs, date and time properties, and key properties.

Title	Description
Basic Development Guide	Provides information about working with records, dimensions, and basic search features.
Advanced Development Guide	Covers such topics as Endeca Query Language (EQL), record filters, bulk export, spelling correction, phrasing, relevance ranking, and dynamic business rules.
MDEX Engine Installation Guide	Provides a brief overview of the Endeca MDEX Engine, details installation procedures, and describes how to configure the licensing keys for the Language Pack. Covers both Windows and Linux/UNIX system requirements and installation procedures.
MDEX Engine Migration Guide	Provides information on migrating from previous versions of Endeca software.
Partial Updates Guide	A guide to preparing and running partial updates in your Endeca application.
Performance Tuning Guide	Provides guidelines on monitoring and tuning the performance of the Endeca MDEX Engine. Contains tips on resolving associated operational issues.
Web Services and XQuery Developer's Guide	Describes how to use Web services and XQuery for Endeca. Web services and XQuery for Endeca provides Endeca application developers with a flexible, extensible, and standards-compliant query processing solution.
MDEX Engine Release Notes	Details the changes specific to this release, including bug fixes and new features.

Presentation API documentation

The following table lists the documentation for the Presentation API package.

Title	Description
Presentation API for Java Reference (Javadoc)	The Java reference documentation for the Endeca Presentation, Analytics, and Charting APIs.
Presentation API for .NET Reference	The .NET reference documentation for the Endeca Presentation, Analytics, and Charting APIs.
Logging API for Java Reference (Javadoc)	The Java reference documentation for the Endeca Logging API.
Logging API for .NET Reference	The .NET reference documentation for the Endeca Logging API.

Platform Services documentation

The following table lists the documentation related to the Platform Services package.

Title	Description
Content Adapter Developer's Guide	Describes how to write Java manipulators and content adapters using the Endeca Content Adapter Development Kit.
Control System Guide	Provides information on using the Endeca Control System, including communicating with the JCD service and running control scripts.
EAC Guide	Describes the tasks involved in managing implementations using the Endeca Application Controller.
Forge Guide	The essential reference for developers of the back-end of Endeca applications (the instance configuration), including Forge pipeline-related tasks.
Log Server and Report Generator Guide	Describes how to configure and run the Endeca Log Server and the Report Generator.
Platform Services Installation Guide	Describes how to install the Endeca Platform Services software, the Endeca Document Conversion Module, and the optional Corda software. Covers both Windows and Linux/UNIX system requirements and installation procedures.
Security Guide	Describes how to implement user authentication and how to structure your data to limit access to only those users with the correct permissions.
Data Foundry Expression Reference	Describes the Data Foundry expression language, used in record manipulators in Developer Studio.
Developer Studio Help	Help (including context-sensitive help) for using Endeca Developer Studio to define all aspects of your instance configuration, including properties, dimensions, and pipelines.
Forge API Guide for Perl	Describes the classes and methods you can incorporate into Perl manipulators in Developer Studio. You can use Perl manipulators in pipelines to manipulate records.
XML Reference	Describes the XML elements contained in the XML and DTD files of the Endeca Information Transformation Layer.
API reference documentation (Javadoc and .NET API reference)	The reference documentation for the Endeca Presentation, Logging, Analytics, and Charting APIs.

Endeca Workbench documentation

The following table lists the documentation related to the Endeca Workbench package.

Title	Description
Endeca Workbench Administrator's Guide	The essential guide for administrators of Endeca implementations and application developers who maintain and customize Workbench instances.
Endeca Workbench User's Guide	The essential guide for business users of Endeca Workbench. Describes enhancements business users can make to Endeca

Title	Description
	implementations with a focus on working with dynamic business rules, search configuration, and reports.
Endeca Workbench Installation Guide	Describes how to install the Endeca Workbench software. Covers both Windows and Linux/UNIX system requirements and installation procedures.
Endeca Workbench Migration Guide	Provides information on migrating from previous versions of Endeca software.
Endeca Workbench Help	Help (including context-sensitive help) for using Endeca Workbench to perform business-user tasks and administer an Endeca implementation. There are versions for each Workbench edition.
Endeca Workbench Release Notes	Details the changes specific to this release, including bug fixes and new features.

Content Assembler API documentation

The following table lists the documentation related to the Content Assembler API, used in conjunction with the Page Builder component of Endeca Workbench.

Title	Description
Page Builder Developer's Guide	Describes the process of developing templates and other supporting tasks to enable content administrators to configure dynamic landing pages using the Endeca Page Builder. Also describes extending Page Builder functionality with community editors.
Content Assembler API Developer's Guide	Describes the process of developing applications with cartridges (for use with the Endeca Page Builder), including usage of the Content Assembler API and an overview of the reference applications. Also describes extending Content Assembler functionality with community tag handlers. There are versions for Java and .NET.
API reference documentation (Javadoc and .NET API reference)	The reference documentation for the Endeca Content Assembler APIs.
Page Builder Editor API reference	The reference documentation for the Page Builder Editor API, part of the Page Builder Editor SDK.
Content Assembler Release Notes	Details the changes specific to this release, including bug fixes and new features.

Content Acquisition System (CAS) documentation

The following table lists the documentation related to the Content Acquisition System (CAS) package.

Title	Description
CAS Console Help	Describes the tasks involved in managing various data sources including file systems, Documentum repositories, and other CMS repositories using the CAS Console for Endeca Workbench.
CAS Developer's Guide	Provides an overview of the Endeca Content Acquisition System, including the Endeca CAS Server, the Component Instance Manager, and the Record Store. The guide also explains how to create a Forge pipeline that utilizes the source data gathered from file system and CMS crawls.
CAS Extension API Guide	Describes how to implement, test, and package CAS extensions using the CAS Extension API.
CAS Installation Guide	Describes how to install the Endeca CAS software. Covers both Windows and Linux/UNIX system requirements and installation procedures.
CAS Migration Guide	Describes the major migration tasks for the suite of CAS components.
CAS API Guide	Provides reference information about the Endeca CAS Server API, the Component Instance Manager API, and the Record Store API.
CMS Connector Guides	Describe the tasks involved in enabling and configuring the various CMS connectors for use with the CAS Server. These guides are available only from the Product Downloads section of the Endeca Developer Network (EDeN).
Web Crawler Guide	Describes the major tasks involved in configuring the Endeca Web Crawler and using it to run crawls that gather source data from Web sites.
CAS Release Announcement	Describes the major new features in this release.
CAS Release Notes	Details the changes specific to this release, including bug fixes and new features.

Rapid Application Development (RAD) Toolkit documentation

The following table lists the documentation related to the Rapid Application Development (RAD) Toolkit for ASP.NET.

Title	Description
RAD Toolkit Developer's Guide	The essential guide for developers of the front-end of Endeca applications (primarily API-related tasks). Also includes information about installation tasks.
RAD Toolkit Release Announcement	Describes the major new features in this release.

Title	Description
RAD Toolkit Release Notes	Details the changes specific to this release, including bug fixes and new features.
API reference documentation (.NET API reference)	The reference documentation for the Endeca RAD API. See also the Input Types and Output Types diagrams for additional information about the API.

Documentation for other packages

The following table lists the documentation related to other Endeca packages.

Deployment Template

Title	Description
Deployment Template Usage Guide	Describes the Deployment Template directories and script functionality, and identifies touch-points where developers may need to configure or extend the template for their projects.
Release Notes (CHANGES)	Details the changes specific to this release, including bug fixes and new features.

Developer Studio

Title	Description
Developer Studio Installation Guide	Provides an overview of Developer Studio and describes system requirements and installation procedures.
Developer Studio Help	Help (including context-sensitive help) for using Endeca Developer Studio to define all aspects of your instance configuration, including properties, dimensions, and pipelines.
Data Foundry Expression Reference	Describes the Data Foundry expression language, used in record manipulators in Developer Studio.
Forge API Guide for Perl	Describes the classes and methods you can incorporate into Perl manipulators in Developer Studio. You can use Perl manipulators in pipelines to manipulate records.
XML Reference	Describes the XML elements contained in the XML and DTD files of the Endeca Information Transformation Layer.

Search Engine Optimization Module

Title	Description
Sitemap Generator Developer's Guide	Describes the Endeca Sitemap Generator and provides instructions for using it to generate sitemaps for an Endeca application.
URL Optimization API Developer's Guide	Describes the major tasks involved in developing an application that utilizes the Endeca URL Optimization API. There are

Title	Description
	versions for Java, the Presentation API for ASP.NET, and the RAD Toolkit for ASP.NET.
API reference documentation (Javadoc and .NET API reference)	The reference documentation for the URL Optimization APIs.

Appendix B

Endeca Environment Variables and Port Usage

This section lists all the environment variables and ports used by the Endeca software. Depending on which components you have installed, not all of them may apply to your implementation.

Endeca environment variables

The Endeca installation programs create several environment variables.

For each variable, the first value listed is the path if you accept the default installation path on Windows (under `C:\Endeca\`*`product`*) and use a per-machine installation. The default paths for a per-user installation will be rooted in the `%USERPROFILE%` directory.

The second value is the path within your installation directory on UNIX. For example, if you install Endeca to `/usr/local/`, the full path of `ENDECA_ROOT` would be `/usr/local/endeca/Platform¬Services/`*`version`* in your environment.

In addition to creating the variables below, the installation may add Endeca directories to the `PATH` variable.

Note: For the MDEX Engine installation, environment and `PATH` variables are set by running the `mdex_setup` scripts provided by the installation. See the *MDEX Engine Installation Guide* for more information.

MDEX Engine variables

The following variable is used by the MDEX Engine:

Variable	Description	Default value
`ENDECA_MDEX_ROOT`	Specifies the path of the MDEX Engine root directory.	• `C:\Endeca\MDEX\`*`version`* • `endeca/MDEX/`*`version`*

Platform Services variables

The following variables are used by the Platform Services:

Variable	Description	Default value
ENDECA_ROOT	Specifies the path of the Platform Services root directory.	• C:\Endeca\PlatformSer¬ vices*version* • endeca/PlatformServices/*ver¬ sion*
ENDECA_REFERENCE_DIR	Specifies the path of the directory that contains the Endeca reference implementations, such as the sample wine project and the JSP and .NET UI references.	• C:\Endeca\PlatformSer¬ vices\reference • endeca/PlatformServices/ref¬ erence
ENDECA_CONF	Specifies the path of the workspace directory for the Endeca HTTP service, which contains configuration files, logs, and temporary storage directories.	• C:\Endeca\PlatformSer¬ vices\workspace • endeca/PlatformSer¬ vices/workspace
PERLLIB	Specifies the path of the perl root directory and its directory of libraries.	• %ENDECA_ROOT%\perl and %ENDE¬ CA_ROOT%\perl\5.8.3\lib • $ENDECA_ROOT/lib/perl:$ENDE¬ CA_ROOT/lib/perl/Control:$EN¬ DECA_ROOT/perl/lib:$ENDE¬ CA_ROOT/perl/lib/site_perl
PERL5LIB	Same as the PERLLIB variable.	Same as the PERLLIB variable.
UnixUtils	Specifies the path of the utilities directory, which contains Windows versions of some UNIX common utilities.	• %ENDECA_ROOT%\utilities • not available on UNIX

Endeca Workbench variables

The following variables are used by the Endeca Workbench:

Variable	Description	Default value
ENDE¬ CA_TOOLS_ROOT	Specifies the path of the Endeca Workbench root directory.	• C:\Endeca\Workbench*version* • endeca/Workbench/*version*
ENDE¬ CA_TOOLS_CONF	Specifies the path of the workspace directory for the Endeca Tools Service, which	• C:\Endeca\Workbench\workspace • endeca/Workbench/workspace

Variable	Description	Default value
	contains configuration files, logs, and temporary storage directories.	

Other variables

Other variables used by Endeca include the following:

Variable	Description	Default value
`ENDECA_PROJECT_DIR`	Specifies the path of the deployed application. This variable is set and used by the Endeca Deployment Template.	Value is taken from user input at installation time.
`ENDECA_PROJECT_NAME`	Specifies the project name that is used, for example, as the JCD job prefix for jobs defined in the project's Job Control Daemon. This variable is set and used by the Endeca Deployment Template.	Value is taken from user input at installation time.

Endeca ports

This topic describes the ports used by the Endeca packages and their default port numbers.

You can replace any of the default port numbers with numbers of your own, as long as they do not conflict with an existing port on your machine. Port numbers can be no larger than `32767`.

Service ports

Port	Default
Endeca Tools Service port	`8006`
Endeca Tools Service SSL port	`8446`
Endeca Tools Service shutdown port	`8084`
CAS Service port	`8500`
CAS Service shutdown port	`8506`
Endeca HTTP Service port	`8888`
Endeca HTTP Service SSL port	`8443`
Endeca HTTP Service shutdown port	`8090`
Endeca Control System JCD port	`8088`

Port	Default
Note: The JCD is deprecated.	

Deployment Template ports

These are the port numbers suggested by the Deployment Template installation, but you can specify any other port when you deploy your application.

Port	Default
Dgraph1 user query port	`15000`
Dgraph2 user query port	`15001`
Agraph1 user query port (Agraph deployments only)	`14000`
Agraph2 user query port (Agraph deployments only)	`14001`
Forge server (Agraph deployments with Parallel Forge only)	`14099`
Endeca Logging and Reporting Server port **Note:** The Logging Server port number can be no larger than `32767`.	`15010`

Reference implementation ports

These port numbers are used in the configuration files that ship with the reference implementation (`sample_wine_data`).

Port	Default
Endeca MDEX Engine user query port	`8000`
Endeca Logging and Reporting Server port **Note:** The Logging Server port number can be no larger than `32767`. In the JSP reference implementation, the default Logging server port number is larger by 2 than the corresponding Dgraph port number. For example, for the Dgraph port 15000, the default port for the Logging Server in the reference implementation is 15002. For the Dgraph port 15001, the default port for the Logging Server in the reference implementation is 15003. (This assumes that the Logging Server is running on the same host as the MDEX Engine.)	`8002`

Index

P

R

S

T

V