

# Upgrading to Oracle's JD Edwards EnterpriseOne 8.9 Unicode on the IBM System i platform



*Randy Erickson  
Software Engineer  
System i ERP Development*

*Gerrie Fisk  
Senior System i IT Specialist  
IBM Oracle ICC*

*June 2007*



## Table of contents

Change history .....	1
Introduction .....	2
Environment .....	2
JD Edwards EnterpriseOne ERP 8.0 .....	3
Upgrade process .....	3
JD Edwards EnterpriseOne 8.9 (non-Unicode).....	3
JD Edwards EnterpriseOne 8.9 – Unicode conversion.....	7
Relative Performance .....	9
Conclusions.....	10
Resources .....	11
Trademarks and special notices.....	12



## Change history

Version	Date	Editor	Editing description
1.0	06/30/2004	Gerrie Fisk	Original version
2.0	06/15/2007	Patrick Moore	Terminology and document template updated



## Introduction

---

This paper documents the technical experience of upgrading to JD Edwards EnterpriseOne 8.9 non-Unicode on an IBM System i™ platform, followed by upgrading the database from non-Unicode to Unicode. In today's global economy, many businesses are analyzing the requirements for installing a Unicode database because of its capability to handle multiple languages without resorting to multiple database instances. The results include timing the process of the two upgrades as well as tracking the changes in the size of the database. Helpful tips discovered along the way are also reported as well as some resource planning guidelines.

## Environment

---

The lab environment used for the writing of this paper consisted of a System i model 810-2469 (2700 CPW) 2-way running OS/400® V5R2, with an Integrated xSeries™ Server (IXS) model 2892-001 running Windows® 2000 Server SP4. The System i server was installed as an all-in-one configuration which means that the application, database, Web, and deployment servers are all on the same system, with the IXS card for deployment. Because this was a lab environment, only the production environment was installed and measured. There were no custom modifications.

See Table 1 and Table 2 for the hardware and software specifications of the System i server.

System i configuration	
OS level	V5R2
Model/CPW	810-2469, CPW 2700
Processors	2 way - 750 MHz
Ethernet – System i & IXS	1 GB
Memory	16 GB
DASD - 15 17GB drives	228 GB

Table 1 - Hardware specifications

Software:	JD Edwards EnterpriseOne ERP 8.0, SP22_C1	JD Edwards EnterpriseOne 8.9 SP1_B1
PTF level - cumulative	C03161520	C03364520
Hiper SF99519	64	90
Database SF99502	Service Pack 9	Service Pack 11
WebSphere® SF99148	Version 4.0.5, Service Pack 6	Version 4.0.5 Service Pack 6
HTTP Server SF99098	Service Pack 10	Service Pack 10

Table 2 - Software specifications

## JD Edwards EnterpriseOne ERP 8.0

---

The systems were installed with JD Edwards EnterpriseOne ERP 8.0 SP22\_C1, following the application installation guide procedures. A sample customer database was used to establish the relative amount of disk storage required. Library and directory sizes were collected for setting a baseline measurement. The total disk space required for the JD Edwards EnterpriseOne ERP 8.0 installation with one pathcode was 32.9 gigabytes as shown in Table 3. Keep in mind that the lab environment is only a subset of the full database load of a normal business environment with multiple pathcodes.

### Upgrade process

---

When preparing to upgrade to JD Edwards EnterpriseOne 8.9, a review of the JD Edwards EnterpriseOne 8.9 Upgrade Guide for iSeries® – Based Systems manual is necessary. A quote from the Customer Overview chapter states:

“Unicode specifies that the data stored in the data source is in Unicode format. For installs, all data sources default to Unicode. For upgrades, all data sources but Business Data and Control Tables default to Unicode, which may be converted to Unicode after the upgrade. Unicode is required for Central Objects data sources and, for UDB, any other data sources using the same database as Central Objects.”

“Please note that the code page still must be set to the correct value even though the data sources may all be Unicode. The Unicode flag only indicates what column type is used to store character data in a Unicode database. For example, for DB2 AS/400, it indicates GRAPHIC/VARGRAPHIC with CCSID 13488 is used.”

The recommended upgrade process, therefore, is a two-step approach: first, upgrade your system to JD Edwards EnterpriseOne 8.9 where all programs and files are automatically Unicode (except for the non-Unicode database such as PRODDTA and PRODCTL). Second, convert the PRODDTA and PRODCTL databases to Unicode.

### JD Edwards EnterpriseOne 8.9 (non-Unicode)

---

The systems were upgraded to JD Edwards EnterpriseOne 8.9, following the JD Edwards EnterpriseOne 8.9 Upgrade Guide for iSeries. Installing the deployment server took four hours as did installing the CDs to upgrade the enterprise server. The JD Edwards EnterpriseOne 8.9 upgrade process, which includes the Workbench and Table Conversions (TCs), also took about four hours. The client/server package build took 9 ¾ hours.

For Table Conversion throughput, the default setting in JDEPLAN for number of threads (queues) performing Table Conversions is “4.” When initially creating the custom plan, you can change this default. Oracle recommends setting it to one or more per processor on the system, depending on the size of your System i server, other workloads, and the network configuration and traffic volumes. In the lab, since our System i server was dedicated to TCs and our dedicated network was one gigabit, we set the threads to “5” on a 2-way, which took the system to 95% CPU utilization.

To modify the threading for TCs, follow the steps in the “Running Table Conversions” section of the JD Edwards EnterpriseOne 8.9 Upgrade Guide for iSeries. A summary of the process is:



1. In the Plan, set Table Conversion throttle to x, where x is the number of threads that you choose for the size of your system, by going to the Advanced Parameters for the Environment. While still logged into JDEPLAN, run application P986130 and set local queue to x. (This local queue setting from the Initial Tasks Workbench updates the queue setting on the enterprise server during the initial plan creation.)
2. Change JOBQ QBATCH on the enterprise system to x.  
CHGJOBQE SBSDB(QBATCH) JOBQ(QBATCH) MAXACT(x)  
(To change the queue setting after the Initial Tasks Workbench has been run for the first plan, you must enter DEP9 and run P986130 to set QBATCH default queue with Max batch jobs =x, because only the first plan modifies the enterprise server.)

The Universal Batch Engine (UBE) jobs (500 or so) never appear to be waiting in the JOBQ so do not look there! To check your progress on the deployment server, drill down in Custom Plan (GH961 - highlight your plan and press Expand) - highlight Table Conversions and press Select. Use the QBE line and look for conversions where status is <60. To check for errors in completed TCs, run P984052 and in conversion status search on >2.

Also, there is a new SQL Package Library parameter in the host jde.ini file:

```
[DB SYSTEM SETTINGS]
```

```
SQL Package Library=1
```

This parameter is shipped as “turned on.” It needs to be on or active (i.e., 2) only during the Table Conversions (TCs) of an upgrade to JD Edwards EnterpriseOne 8.9. The TCs are a special kind of UBE job that all share the same name. If the next UBE uses the previous UBE’s SQLPKG, it will not find the framework that it needs, slowing down the upgrade. After the upgrade to JD Edwards EnterpriseOne 8.9 is completed, remember to turn this setting off (i.e., 1) in the jde.ini file on the enterprise server. In other words, this parameter should be turned off (i.e., 1) or commented out for installs, normal operations, including the subsequent upgrade to Unicode data.

In releases prior to JD Edwards EnterpriseOne 8.9, there were a few files, including F989998 and F989999 (Central Objects), that needed data fields longer than 30,000 characters. The solution was to chain multiple records together to form a single record. The F98BLOB table is used in JD Edwards EnterpriseOne ERP 8.0 and earlier releases to hold this overflow data consisting of the chained records. This solution is informally referred to as the BLOB (binary large object) chaining solution. With JD Edwards EnterpriseOne 8.9, the F98BLOB table no longer exists because JD Edwards EnterpriseOne 8.9 and later releases now use a true large object (LOB). Set LOB=True in the jdbj.ini file sections so that the F98BLOB table is not used.

```
[JDBj-BOOTSTRAP SPEC DATA SOURCE]
```

```
LOB=True
```

```
[JDBj-SPEC DATA SOURCE]
```

```
LOB=True
```

During the upgrade, the Table Conversion engine recreates Central Objects files including F989998 and F989999 in a new LOB data source and automatically converts those records using the BLOB chaining solution into true LOBs.



Note: Run the Display File Field Description command (DSPFFD) for the F989999 file to verify its status. If the field WBJPO has a data type of "HEX" with a field length of 30,000 and the notation "Variable length field", this file is using the BLOB chaining solution and the upgrade to JD Edwards EnterpriseOne 8.9 has not occurred. The operating system sets a true LOB data type to "BLOB", so if the field WBJPO has a data type of "BLOB" and a large field length (larger than 30,000), then this file is using the true LOB solution and the conversion was successful.

Be sure to follow the directions in the JD Edwards EnterpriseOne 8.9 Upgrade Guide for iSeries in the Custom Modifications and Packages chapter to modify the QAQQINI file. The file needs to be duplicated into QUSRSYS and the following entry needs to be added to QUSRSYS's QAQQINI file. Without this entry to the QAQQINI file, anything that fetches more than 250,000 LOBs, such as a server package build, will fail:

```
LOB_LOCATOR_THRESHOLD 10000
```

Another tip for viewing and editing Integrated File System (IFS) files via iSeries Navigator is to drill down >File Systems ->Integrated File Systems ->Root, and right-click to add a Read/Write Share name of Root. Then under Text Conversion, select "Allow Simple Conversion". Add .txt, .log, .ini, .mbr, .sts and .properties extensions for viewing text files in iSeries Navigator.

As shown in Table 3, the total disk space required for JD Edwards EnterpriseOne 8.9 installation with one pathcode was 31.9 gigabytes, which was slightly smaller than JD Edwards EnterpriseOne ERP 8.0. The size of the packages is smaller, while the database grew 25%.



	JD Edwards EnterpriseOne ERP 8.0	JD Edwards EnterpriseOne 8.9 - Non Unicode	JD Edwards EnterpriseOne 8.9 - Unicode
Library & Directory Sizes	Measured in MB	Measured in MB	Measured in MB
B7334SYS - SP22_C1 / B9SYS - SP1_B1	451	215	215
COPD7334 / COPD9 - Central Objects	2,793	3,182	3,182
DD7334 / DD9 - Data Dictionary	75	240	240
OL7334 / OL9 - Object Librarian	110	170	170
OWINSTALL - OW Save File Library	2,837	3,425	3,425
OWJRNL - OneWorld Journal Library	0	1	1
PD7334 / PD9 - Production Pathcode	688	839	899
PD7334DNT - Versions for PD7334	6	0	
PD7334FA / PD9FA - Package Library	3,389	1,176	1,176
PD7334IN / CLARK - Package Library	3,271	1,757	1,757
PRODCTL	120	143	417
PRODDTA	12,121	15,137	20,678
SVM7334 / SV9 - Server Map	6	9	9
SYS7334 / SY9 System Library	97	125	125
Total libraries:	25,964	26,419	32,294
/B7334SYS - /B9SYS	55	67	67
/JDEB7334 - JDEB9 logs	1	1	1
/JDEdwards - /PeopleSoft	129	184	184
/OneWorld packages-2 for ERP 8, 2 for Rel.8.9	4,600	3,800	3,800
/PD7334 - /PD9 specs	2,160	1,470	1,470
Total IFS:	6,945	5,522	5,522
Total disk for one pathcode:	32,909	31,941	37,816

Table 3 - Disk space usage





## JD Edwards EnterpriseOne 8.9 – Unicode conversion

---

The JD Edwards EnterpriseOne 8.9 customer database was converted to Unicode, following the steps in the JD Edwards EnterpriseOne 8.9 Data Conversion PeopleBook.

IBM ships the QCCSID system value set to 65535. Before converting to Unicode, we used this CCSID because it would tell iSeries Access to let characters come through without translation; the system value QCCSID was 65535 and the JD Edwards application user profile's CCSID was 37. Since the lab installation used English only, the system value QCCSID was changed to 37 in order that any user profile establishing a database connection could access the data (e.g., a user profile that uses STRSQL to view the new Unicode sources such as Central Object files, PRODDTA files, etc.). (The Unicode CCSID is 14388.) It may be preferable to change CCSID at the user profile level, not system-wide.

**Note:** Environments requiring multiple language support may have different requirements and is outside the scope of this paper.

To check for Unicode, run a display file field definition and look for field types of either CHAR or GRAPHIC. A non-Unicode database has CHAR field definitions. A Unicode database has GRAPHIC field definitions. For example: DSPFFD PRODCTL/F0002.

To view logs from within Windows, go into iSeries Navigator and navigate as follows: ->Network ->Servers ->TCPIP ->NetServer ->Advanced ->Next ->Start ->CCSID ->Browse, and change the CCSID from 0 to 437 and then bounce NetServer.

The steps to convert to Unicode consist of a prerequisite step, three main steps, and a post-requisite step:

The prerequisite step runs an SQL query which lists the obsolete tables in Object Librarian Table F9860. Then delete these tables from the database prior to converting. (This cleanup step is designed to save disk space because the subsequent Unicode conversion actually ignores these obsolete tables and any other extra customer files in PRODDTA or PRODCTL). Following is one method to delete these tables from the database prior to converting.

In the lab, there were 68 obsolete physical files (PFs) in PRODDTA and none in PRODCTL. Run WRKOBJPDM PRODDTA to display the list of all files. Then, to look at the list of 68 files in SQL using iSeries Navigator, enter the SQL command statement:

```
select SIOBNM from OL9.F9860 where SISY = '89' and SIFUNO = 'TBLE';
```

Read down the SQL list and delete the logical files (LFs) followed by the physical file (PF) in the green-screen session.

The three main steps after the prerequisite are as follows:

1. Run a local UBE P93091 on the deployment server to build the Unicode Data Conversion master table from the system and server map Data Source master tables which are F98611 in SY9 and SVM9.
2. Run the data source/table analyzer and conversion program which creates the XML script that defines all the data sources and tables to be converted. This script is the input to the conversion executable:

C:/B9/script/UniDtaConv.xml

Evaluate this script for completeness and accuracy because it must run successfully or start completely over again. Ensure that all the data sources to be converted are listed and no extraneous ones have inadvertently been added. The analyzer may automatically add dependent data sources it thinks you need. In the lab, the XML script included some local databases on the deployment server which caused the script to fail until they were removed.

**Note:** Although not recommended, the Conversion Executable step was abnormally ended in the lab. The chapter entitled “Error Handling” in the JD Edwards EnterpriseOne 8.9 Data Conversion People-Book was used to assist in the restart. The tables with the suffix “\_UNICODE” did not drop as directed. An IPL did not fix the problem. It was necessary to run a RCLSTG \*DBXREF to clean up the cross-reference files. The RCLSTG completed in six minutes, after going to restricted state.

3. Run the conversion executable after all users have signed off and all applications are down. It converts the metadata from non-Unicode to Unicode and also translates the data from their respective language IDs to a Unicode ID (CCSID).

C: B9/system/Bin32/Unicode.exe

Check the error log:

C:/B9/script/UniDtaConv\_Log.xml

The post-requisite step follows after the conversion executable finishes. At this point, all the physical files (PFs) are Unicode and there are no logical files (LFs). They need to be recreated in PRODDTA and PRODDCTL. Start JD Edwards EnterpriseOne services and log in to DEP9 on the deployment server to run UBE R9698713. It runs locally to rebuild the logical files. We chose the smaller PRODDCTL first, then PRODDTA. When rebuilding of the logical files is finished, stop JD Edwards EnterpriseOne services, delete all \*SQLPKG (except for QZDAPKG, QSQLPKG2, and QSQXDPKG) and restart JD Edwards EnterpriseOne services.

During the Unicode conversion we monitored the temporary space utilized via WRKSYSSTS to determine how much workspace the file conversion required. Temporary space grew to the relatively modest size of around three gigabytes. The largest file in the lab’s database is F0911 at 1.5 gigabytes. The conversion used around 44% CPU with five QZDASOINIT jobs running.

Unicode conversion times took 3:07 (hours:minutes) to run the conversion script and 1:21 to rebuild the logical files for a total of about 4 ½ hours. Although all other libraries and directories remained unchanged (see Figure 3), the database increased 38% over the JD Edwards EnterpriseOne 8.9 non-Unicode database as shown in Table 4. The percentage increase in size for a Unicode database is dependent on how much text exists relative to the database as a whole. For example, there are VARCHAR fields which have variable lengths for text content. If the description fields are very long in general, the % uplift for Unicode database sizing could be larger.



One pathcode (in MB)	JD Edwards EnterpriseOne ERP 8.0	JD Edwards EnterpriseOne 8.9 non-Unicode	JD Edwards EnterpriseOne 8.9 Unicode	% growth
Total disk usage including database:	32,909	31,941	37,816	
% growth ERP 8.0 to non-Unicode:		-3%		
% growth non-Unicode to Unicode:			18%	
Total % growth ERP 8 to Unicode:				15%
Database only:				
PRODCTL	120	143	417	
PRODDTA	12,121	15,137	20,678	
Total database	12,241	15,280	21,095	
% database growth ERP 8 to non-Unicode:	25%			
% database growth JD Edwards EnterpriseOne 8.9 non-Unicode to Unicode:			38%	
Total % database growth JD Edwards EnterpriseOne ERP 8.0 to JD Edwards EnterpriseOne 8.9 Unicode:				72%

Table 4 - Disk space growth

## Relative Performance

In a separate series of lab tests designed to tune and analyze relative performance, we have executed JD Edwards EnterpriseOne 8.9 non-Unicode and Unicode standard JD Edwards EnterpriseOne performance tests and have compared them with our previous tests running JD Edwards EnterpriseOne Xe (comparable to JD Edwards EnterpriseOne ERP 8.0).

For planning purposes, Table 5 lists the relative resources that may be required when upgrading from JD Edwards EnterpriseOne ERP 8.0 to JD Edwards EnterpriseOne 8.9. (The ongoing testing in the lab continues to yield improvements.)

Upgrade planning for JD Edwards EnterpriseOne 8.9		
	CPU % Uplift	Memory % Uplift
V3T	22 %	10 %
JAS	70 %	no change
AIO	38 %	6 %

Table 5 - CPU and memory uplifts

Virtual 3 Tier (V3T) indicates that a System i platform with JD Edwards EnterpriseOne Xe or JD Edwards EnterpriseOne ERP 8.0 application and database servers will require 22% more CPU and 10% more



memory when upgrading to JD Edwards EnterpriseOne 8.9. Java™ Application Server (JAS) shows that the HTML front-end Web server (a separate standalone System i server with WebSphere and HTTP servers) requires 70% more CPU and the same amount of memory. All-in-one (AIO) uplift means that a business whose System i server runs application, database, and WebSphere servers all on the same system need to plan on the need for 38% more CPU and 6% more memory. The AIO percentages reflect the fact that the enterprise (V3T) workload is weighted slightly more than the JAS when the two workloads are combined.

The performance differences between non-Unicode and Unicode were slight so Table 5 is used for both. (Tests run with Unicode database used 4% less CPU than non-Unicode.)

**Note:** These guidelines are for planning purposes only. Individual results may vary because of the variety and variability in the installations.

## Conclusions

---

In the lab, the upgrade to JD Edwards EnterpriseOne 8.9 took approximately four hours, an additional four hours to run the Workbench, and 9¾ hours for a full client/server package build. JD Edwards EnterpriseOne 8.9 required only slightly more disk than JD Edwards EnterpriseOne ERP 8.0 overall, although the database portion (PRODDTA/PRODCTL libraries) grew 25%.

The upgrade from non-Unicode to Unicode took 4 ½ hours. The Unicode version of the database (PRODDTA/PRODCTL libraries) caused 38% uplift in disk space required. During the conversion, the temporary space growth required was equal to approximately double the size of the largest file.

As measured in the lab, the total database growth (PRODDTA/PRODCTL libraries) when upgrading from JD Edwards EnterpriseOne ERP8.0 to JD Edwards EnterpriseOne 8.9 Unicode was 72%. Since this percentage is dependent on the size of the text in the variable-length fields, results may vary. This information is available as an example of database growth and does not include other factors, such as custom modifications, versions, and PRINTQUEUE.

Although system resources have increased with the introduction of Unicode, the benefits of being able to run multiple code pages and multiple languages, while not having to separate the database into multiple instances are significant. This paper will help you get there with database sizing examples, a summary of the upgrade process, resource requirements, and helpful tips along the way.



## Resources

---

These Web sites provide useful references to supplement the information contained in this document:

- IBM System i home page  
<http://www.ibm.com/systems/i/>
- IBM eServer iSeries [System i] Information Center  
<http://publib.boulder.ibm.com/series/>
- IBM System i Performance and Tuning Tips for Oracle's JD Edwards EnterpriseOne 8.12 and WebSphere 6.0  
<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100985>
- IBM System i5 520 with POWER5 Performance in a JD Edwards EnterpriseOne Mixed Workload Environment  
<http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100511>
- IBM Hardware Sizing Questionnaires  
<http://ibm.com/erp/sizing>
- IBM Publications Center  
[www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US](http://www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US)
- IBM Redbooks  
[www.redbooks.ibm.com/](http://www.redbooks.ibm.com/)
- IBM / Oracle InfoAPAR for JD Edwards applications  
[http://www.ibm.com/servers/enable/site/events/jde\\_links.html](http://www.ibm.com/servers/enable/site/events/jde_links.html)



## Trademarks and special notices

---

© Copyright. IBM Corporation 1994-2007. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, iSeries, OS/400, System i, WebSphere, and xSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.



Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.