

# Migrating Active Server Pages Applications from Windows NT/2000 to UNIX/Linux

---

## Abstract

This document will provide a basic overview of what is involved in preparing for, and performing a migration of an ASP based Web application from a Microsoft Windows environment to a Unix or Linux based environment. This should be used as a guide for researching what is involved in deploying ASP Applications in a cross-platform environment, as well as a resource for common issues and potential work-arounds or solutions, during the actual deployment.

---



901 San Antonio Road  
Palo Alto, CA 94303  
1 (800) 786.7638  
Sun Microsystems, Inc.  
1.512.434.1511

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Solaris, and Solstice PC-Admin are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Intrinsyc and J-Integra are trademarks registered trademarks, or service marks of Intrinsyc Software, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**RESTRICTED RIGHTS:** Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

# Table of Contents

---

<b>Introduction</b>	<b>4</b>
<b>Migrating ASP Applications</b>	<b>5</b>
<i>STEP 1: Moving the ASP Application</i>	5
<i>STEP 2: Identifying ASP Application Dependencies</i>	5
<i>STEP 3: Selecting a Strategy for Supporting Dependencies</i>	6
<i>STEP 4: Identifying Database Connectivity Issues</i>	8
<i>STEP 5: Reviewing the ASP Code</i>	11
<b>Conclusion</b>	<b>15</b>

## Introduction

---

Migrating ASP applications from Microsoft Windows NT/2000 to UNIX or Linux requires making the ASP application and its dependencies portable. Once completed, though, the application can be moved to various UNIX or Linux environments (or even back to Windows NT/2000) with minimal effort.

Because planning and preparation are the key elements to a successful migration project, this document explains the process of “portablizing” an ASP application. The time required to migrate an application will depend heavily on the quality of the ASP code and the application dependencies (e.g., custom objects).

Once complete, we suggest reviewing our White Paper “Developing Cross-Platform ASP Applications” before proceeding with future ASP development projects.

# Migrating ASP Applications

---

One of the most common tasks when working with Sun Chili!Soft ASP is to port an existing ASP application from Windows NT/2000 to UNIX/Linux. This task can range from extremely simple to fairly difficult. The difficulty of porting your application can be determined by doing a little bit of research before you begin.

## STEP 1: Moving the ASP Application

To perform the steps below, it's best to transfer the entire ASP application directory structure, and asp code, to the target UNIX/Linux platform first. UNIX/Linux provides a wide variety of tools that can make the research tasks listed below, a lot easier. This is actually a very easy process. All you need to do is FTP the files from the existing Windows machine, to the destination server. If you happen to be lucky enough to be on a Local Area Network with a SAMBA server running, you can simply copy the directory structure of your ASP application over to the new server.

If you cannot FTP or transfer the files via Samba, then you will need to transfer the ASP application to a CD first, and then bring it over to the destination server that way. The one important thing to remember when transferring files from Windows to UNIX/Linux, is that the "/" and "\" (forward slash and back slash) used in path descriptions, does make a difference on Unix. This is normally not an issue, but it is something to look out for. If you use any of the methods above, the programs will take care of this difference for you. If you burn the files to a CD and then move them over, you may have to watch how the files are being written to the UNIX/Linux server.

## STEP 2: Identify ASP Application Dependencies

Probably the most critical part in learning the degree of portability for your ASP application is to identify the application's dependencies. A very quick way to determine what dependencies your ASP application has on COM objects, is to search the entire asp code set for the string:

`Server.CreateObject`

On Unix/Linux, you can use the following syntax to search an entire ASP Application. Run this command from the root of the application.

```
#> find . -name '*.asp' | xargs grep -i 'server.createobject'
```

This syntax is used to instantiate all COM objects in your asp code. The resulting list will provide you with information about all of the COM objects used in your ASP application. You can quickly determine if these objects

are already available on UNIX/Linux or if you will need to find an alternative solution to provide the same functionality.

*NOTE: Sun Chili!Soft ASP ships with a large set of the most commonly used COM objects. Please see [www.chilisoft.com](http://www.chilisoft.com) for a complete listing of included components.*

### **STEP 3: Select Strategy for Supporting Application Components**

Once you have your list of objects, you will need to determine if all the components are supported on the target operating system. There are three core strategies available:

#### **1. Determine if Sun Chili!Soft ASP includes your object**

Below is a list of COM objects that are included with Sun Chili!Soft ASP, and their `Server.CreateObject` syntax:

Dictionary object `Server.CreateObject("Scripting.Dictionary")`  
FileSystemObject object `Server.CreateObject("Scripting.FileSystemObject")`  
Browscap object `Server.CreateObject("MSWC.BrowserType")`  
AdRotator object `Server.CreateObject("MSWC.AdRotator")`  
ContentRotator object `Server.CreateObject("MSWC.ContentRotator")`  
Link object `Server.CreateObject("MSWC.NextLink")`  
Tools object `Server.CreateObject("MSWC.Tools")`  
Counters object `Server.CreateObject("MSWC.Counters")`  
MyInfo object `Server.CreateObject("MSWC.MyInfo")`  
ADO Connection object `Server.CreateObject("ADODB.Connection")`  
ADO Recordset object `Server.CreateObject("ADODB.Recordset")`  
ADO Command object `Server.CreateObject("ADODB.Command")`  
ADO Errors object `Server.CreateObject("ADODB.Errors")`  
ADO Parameter object `Server.CreateObject("ADODB.Parameter")`  
ADO ConnectionPool object `Server.CreateObject("ADODB.ConnectionPool")`  
ADOX Index object `Server.CreateObject("ADOX.Index")`  
ADOX Table object `Server.CreateObject("ADOX.Table")`  
ADOX Column object `Server.CreateObject("ADOX.Column")`  
ADOX Key object `Server.CreateObject("ADOX.Key")`  
ADOX Group object `Server.CreateObject("ADOX.Group")`  
ADOX User object `Server.CreateObject("ADOX.User")`  
ADOX Catalog object `Server.CreateObject("ADOX.Catalog")`  
SMTP Mail object `Server.Createobject("CDONTS.NewMail")`  
POP3 object `Server.Createobject("Chili.Pop3")`  
File Upload object `Server.Createobject("Chili.Upload")`  
COM to Java Bridge `Server.Createobject("Chili.Bean")`

## **2. Determine if a functionally equivalent Java component is available via an internal repository or third-party vendor**

Sun Chili!Soft ASP includes a COM-to-Java bridge referred to as the Chili!Bean. This functionality allows you to script directly to any Java class file with a public constructor. This allows you to take advantage of the many existing Java solutions available from the Internet today. If you work in a large organization, chances are a Java component repository already exists – somewhere.

Below is a small listing of third party vendors that offer solutions or JavaBean repositories for you to search.

<http://www.jars.com>

<http://www.javaworld.com>

<http://softwaredev.earthweb.com/java>

<http://www.jscape.com>

<http://www.componentsource.com/java>

<http://www.flashline.com>

<http://coldjava.hypermat.net>

<http://www.jspsmart.com>

## **3. Create a “component repository” for custom objects using a Windows NT/2000 server**

If you are unable to find your object within Sun Chili!Soft ASP or from the third-party Java component marketplace (or if you’re using a custom component written in Visual Basic), one strategy is to store these components on a single Windows NT/2000 server until they can be re-written in Java.

We suggest the J-Integra Java-to-COM bridge from Intrinsic software. This product will allow you to connect to existing COM objects that reside on a Windows operating system. This is a great option for developers that cannot, or care not to, port to Java.

For a complete step-by-step tutorial on how to use JIntegra with Sun Chili!Soft ASP, please see the white paper titled “Using Intrinsic's J-Integra Java-COM Bridge with Sun Chili!Soft ASP”.

For complete product information for J-Integra from Intrinsic, go to <http://www.intrinsic.com/>.

If you find a COM object in your list that is not provided by Sun Chili!Soft ASP, and there isn’t an equivalent JavaBean available, you will need to rewrite this component in Java. There are several excellent Java IDE tools on the market, including Sun’s Forte for Java and Borland’s

JBuilder. You can download a free version of Forte for Java, Community Edition, from <http://www.sun.com/forte/ffj/overview.html>.

There are also many consulting firms available that can help you with code conversion, if you do not have these resources available to you directly.

#### **STEP 4: Identifying Database Connectivity Issues**

This step involves looking at the database that your ASP application uses. If you are not connecting to a database, you can move on to the final research step.

Sun Chili!Soft ASP includes ODBC drivers for most major databases, including the two most popular OpenSource databases, MySQL and PostgreSQL. These ODBC drivers are different than the ones that are supplied by Microsoft on Windows, and their functionality may be different in some areas. While all ODBC drivers must conform to the ODBC specification, there are various levels of support for this specification. The most common areas that you may encounter a difference in driver behavior are Stored Procedures and supported CursorTypes.

Sun Chili!Soft ASP includes the ActiveX Data Objects (ADO). The ODBC layer of the ADO objects has been implemented only. What this really means to you, as a developer, is that you cannot use a database connection string that uses OLE DB syntax. Both DSN (Data Source Name) and DSN-less connection strings are supported, however. For a full listing of the correct connection string syntax, you should take a look at the Sun Chili!Soft ASP on-line documentation.

When designing and developing the ASP Application, it's most important to design and deploy against the same database. Doing this one thing, will save you an incredible amount of time and frustration. Trying to make adjustments for SQL syntax differences and ODBC driver differences, after you have the ASP Application completed and working smoothly in one environment, can be extremely frustrating.

We'll take a look at each of the major databases individually, and then provide a general section on how other databases not directly covered, should be dealt with.

#### **Microsoft Access**

The very first thing that you will need to know when migrating an ASP application that connects to an Access database, is that Microsoft Access MDB files, will not run on any operating system except Windows. This isn't a limitation of Sun Chili!Soft ASP, but rather the limits of MS Access.

You have two options when working with MS Access databases:

1. Leave the Access MDB file on a Windows machine and connect to it from the UNIX/Linux operating system. Sun Chili!Soft ASP provides an ODBC Client/Server bridge for doing just this. It's not the most efficient solution, but it will work for light traffic web sites.
2. Port the Access database to another database server that is available on the operating system of your choice. MySQL is the most commonly used database server for this option.

There are many tools available to help you with porting the Access database schema and data to a MySQL database. Popular tools include:

- DBTools (<http://dbtools.vila.bol.com.br>) is a tool to manage MySQL databases. It is currently only available for Windows. Some features:
  - Manage servers, databases, tables, columns, indexes, and users
  - Import wizard to import structure and data from MS Access, MS Excel, Dbase, FoxPro, Paradox, and ODBC Databases.
- Various MySQL Tools (<http://www.mysql.com/downloads/contrib.html>)

### ***Common issues with porting to MySQL***

You'll find that porting to MySQL is actually a fairly straightforward process once you understand it. Some differences do exist however. It's difficult to give a comprehensive list of differences, however the most commonly run into issues are when dealing with table names and datatypes used for the schema itself.

#### **Table names**

- MySQL does not allow spaces in the table name.
- Some table names in MS Access may be considered keywords in MySQL. Get a good book on MySQL, or checkout the MySQL web site, to get a better understanding of keywords and what is allowed in MySQL.

#### **Datatypes**

- All databases seem to have a different list of acceptable datatypes beyond the ANSI standards. The most obvious difference between MS Access and MySQL is probably the MEMO datatype in MS Access. This is not available in MySQL, and would most likely be replaced by a longtext datatype.

#### **Schema**

- MS Access provides a feature called "Queries". This is also referred to as "Views" in other databases. Currently, MySQL does not support these and there is not an easy way of transferring these between databases.

## **Microsoft SQL Server**

Microsoft SQL Server is very similar to Microsoft Access when it comes to the cross-platform limits about what operating system it can be run from. Sun Chili!Soft ASP provides a direct ODBC driver for connecting to MS SQL Server databases that are housed on a Microsoft Windows operating system, though.

The limitations that you will run into with MS SQL Server are in a lot of ways, the same as those that we listed in the above section on MS Access, however because of the larger scale implementations of the SQL Server database itself, you will find that most competing database manufacturers such as Oracle, Sybase, and DB2, provide a migration tool for moving between SQL Server and the new database of your choice.

If you are not using MS SQL Server for a medium to high volume application, you may want to consider migrating it to MySQL as well. Please see the MS Access section above for more information about this option.

## **Oracle, Sybase, Informix, DB2**

All of these databases are designed to work in an enterprise environment that takes cross platform development and heterogeneous networks into consideration. Sun Chili!Soft ASP includes ODBC drivers for all of these databases. You will not be required to migrate any data or make schema changes to take advantage of these databases in the new ASP application.

Please refer to the ASP Documentation section on databases for setting up and connecting to these database types.

## **Other ODBC Compliant databases**

Please refer to the Sun Chili!Soft ASP documentation for a complete list of ODBC compliant databases for which an ODBC driver is included. For any other ODBC compliant database that is not listed, but for which you have a UNIX or Linux ODBC driver, it is usually possible to make them work within the Sun Chili!Soft ASP environment. Please contact the Sun Chili!Soft ASP product technical support group for information on how to research this ability. They can be reached through your local Sun technical support representative.

## STEP 5: Reviewing the ASP Code

### Case Sensitivity

Case sensitivity is a major issue when moving an asp application between Windows and UNIX. Depending on how disciplined your developers were when they wrote the original code, you could be looking at anything from no work at all, to quite a bit of testing and filename changes. One thing that may help, but is not always required, is a shell script called "tolower". The code for this script is included below.

Syntax: `#> tolower <path to directory>`

This script will lowercase all files and directories at and below the path that you give it. So.....

`tolower /home/iplanet/docs`

will lowercase all files in /home/iplanet/docs and everything below it.

```
##### Begin Code #####
#!/bin/sh

if [ $# -ne 1 ]; then
echo "Usage: $0 <directory path>"
fi

if [ ! -d $1 ]; then
echo "$0: The parameter must specify a valid start
directory."
fi

IFS='
'for each in `find $1`; do
changed=`echo $each | tr '[A-Z]' '[a-z]`
if [ $each != $changed ]; then
mv $each $changed
fi
done

##### End Code #####
```

## ASP version

Microsoft ASP is currently at version 3.0. The differences between ASP 2.0 and 3.0 are very minor, however, there are a couple of functions that were introduced that Sun Chili!Soft ASP currently does not support. To determine if these functions are used within your ASP application, you will want to search your asp code for the syntax of "Server.Execute" and "Server.Transfer".

On UNIX/Linux, you can use the following syntax to search an entire ASP application. Run this command from the root of the application.

```
#> find . -name '*.asp' | xargs grep -i 'server.execute'
```

and

```
#> find . -name '*.asp' | xargs grep -i 'server.transfer'
```

Both of these features will be added to a future release of Sun Chili!Soft ASP, however, in the meantime, you will need to recode these sections.

For almost all server.execute lines, you can replace them with an HTML <include> statement and get the same results.

For the server.transfer code sections, you can try using a response.redirect. This causes the browser to do the redirect instead of the server, which is a little bit more noticeable by the end user, but should serve the same purpose.

## Scripting Engine Version

Sun Chili!Soft ASP currently ships with VBScript and JScript engines included as part of the product. However these engines are currently written to the 3.2 the specification. This may have an impact on the porting of your ASP application. At this time, there are no major commercial development tools on the market that generate asp code using anything higher than the 3.2 specification, so if your application is written using any of the major tools, then you should not have an issue. If you have written the application from scratch, then you will want to perform the tests below to determine what the impact may be on the migration process.

You will want to search your asp code for some specific syntax that is indicative of VBScript 5.x.

Search for the syntax of "End With" and "RegExp". On UNIX/Linux, you can use the following syntax, when run from the Application root, to search the entire application code base.

```
#> find . -name '*.asp' | xargs grep -i 'End With'
```

and





## Conclusion

---

Migrating ASP applications from Microsoft Windows NT/2000 to UNIX/Linux involves an understanding of the application's dependencies, differences between the Windows operating systems and the UNIX/Linux operating environments, and database limitations. With a proper strategy in place and solid planning, ASP applications can be efficiently migrated (a.k.a. made portable) to a variety of platforms.

### About ASP

ASP is an open, compile-free Web application environment that combines scripting, HTML custom server components, and database publishing to create dynamic Web-based business applications. With ASP, developers can build browser-independent Web solutions and publish databases to the Web using tools such as Microsoft's FrontPage™, Macromedia™ Dreamweaver UltraDev, or Adobe™ GoLive!.

### About Sun Chili!Soft ASP Software

Sun Chili!Soft ASP is the functional equivalent of Microsoft's ASP. While Microsoft ASP is available exclusively on Microsoft's Internet Information Server (IIS) web server on Windows NT/2000, Sun Chili!Soft ASP enables ASP on other leading Web servers and operating systems. Sun Chili!Soft ASP is supported by industry leading Web development tools such as Microsoft FrontPage, Macromedia™ Dreamweaver UltraDev, and Adobe™ GoLive!, and provides ASP functionality to iPlanet, Apache, Zeus and other Web servers.

For more information, please go to <http://www.chilisoft.com>.