



# N1 Grid Service Provisioning System 5.0 User's Guide for the Windows 2000 Plug-In

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

Part No: 819-0412  
December 2004

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 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, docs.sun.com, AnswerBook, AnswerBook2, Java, N1, and Solaris are trademarks or registered trademarks 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.

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.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

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 NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, Java, N1, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



041203@10536



# Contents

---

<b>Preface</b>	<b>7</b>
<b>1 Overview of Windows 2000 Plug-In</b>	<b>11</b>
Purpose of the Windows 2000 Plug-In	11
What the Windows 2000 Plug-In Includes	11
Requirements for Using the Windows 2000 Plug-In	13
Requirements for Deployment Server	13
Requirements for Target Hosts	13
<b>2 Installing and Configuring the Windows 2000 Plug-In</b>	<b>15</b>
Acquiring the Windows 2000 Plug-In	15
Adding the Windows 2000 Plug-In to N1 Grid Service Provisioning System	15
Customizing the Solution for Your Environment	16
<b>3 Using the Windows 2000 Plug-In</b>	<b>17</b>
Capturing and Deploying Files and Applications	17
▼ How to Capture a Windows 2000 File or Application	18
▼ How to Deploy a Windows 2000 File or Application	19
Component Types	20
Windows Applications Component Types	20
IIS Component Types	24
Additional Windows Component Types	28
Microsoft Windows System Services	32
Extended Control Services	32
Extended Control Services for IIS	33

Extended Control Services for Microsoft Windows Services 33

**Index** 35

# Tables

---

<b>TABLE 3-1</b>	COM+ Application Error Conditions	21
<b>TABLE 3-2</b>	COM+ Application Extended Control Procedures	21
<b>TABLE 3-3</b>	COM Component Error Conditions	23
<b>TABLE 3-4</b>	Microsoft Windows Installer File Error Conditions	24
<b>TABLE 3-5</b>	IIS Types Common Error Conditions	25
<b>TABLE 3-6</b>	Registry Keys Error Conditions	29
<b>TABLE 3-7</b>	Registry File Error Conditions	30
<b>TABLE 3-8</b>	Microsoft Windows Batch File Error Conditions	31
<b>TABLE 3-9</b>	Microsoft Windows Scripting Host Script Error Conditions	32
<b>TABLE 3-10</b>	Built-in Procedures for Controlling IIS	33
<b>TABLE 3-11</b>	Built-in Procedures for Controlling Microsoft Windows Services	33



# Preface

---

The N1™ Grid Service Provisioning System 5.0 User's Guide for the Windows 2000 Plug-In contains information about installing, configuring, and using N1 Grid Service Provisioning System to provision Windows COM, COM+, and Internet Information Services (IIS) files and applications.

---

## Who Should Use This Book

The main audience for this book includes system administrators and operators of N1 Grid Service Provisioning System 5.0 software who want to capture and deploy Windows 2000 files and applications with N1 Grid Service Provisioning System software. These users are expected to have the following background:

- Familiar with the N1 Grid Service Provisioning System 5.0 product
- Familiar with standard UNIX® and Microsoft Windows commands and utilities
- Familiar with the general concepts associated with Windows 2000, COM, COM+, and IIS

---

## Before You Read This Book

If you are not already familiar with using the N1 Grid Service Provisioning System software, read the following books:

- *N1 Grid Service Provisioning System 5.0 System Administration Guide*
- *N1 Grid Service Provisioning System 5.0 Operation and Provisioning Guide*
- *N1 Grid Service Provisioning System 5.0 Release Notes*

---

## How This Book Is Organized

[Chapter 1](#) provides an overview of the Windows 2000 Plug-In.

[Chapter 2](#) explains how to install and configure the Windows 2000 Plug-In.

[Chapter 3](#) describes how to use the specific components and plans that are provided with the Windows 2000 Plug-In.

---

## Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

---

**Note** – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

---

---

## Accessing Sun Documentation Online

The docs.sun.com<sup>SM</sup> Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is <http://docs.sun.com>.



---

## Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see “Buy printed documentation” at <http://docs.sun.com>.

---

## Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
<b>AaBbCc123</b>	What you type, contrasted with onscreen computer output	<code>machine_name% <b>su</b></code> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . Perform a <i>patch analysis</i> . Do <i>not</i> save the file. [Note that some emphasized items appear bold online.]

---

## Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

**TABLE P-2** Shell Prompts

<b>Shell</b>	<b>Prompt</b>
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

## Overview of Windows 2000 Plug-In

---

This chapter explains general information about using N1 Grid Service Provisioning System to provision Windows 2000 applications. The chapter contains the following information:

- “Purpose of the Windows 2000 Plug-In” on page 11
- “What the Windows 2000 Plug-In Includes” on page 11
- “Requirements for Using the Windows 2000 Plug-In” on page 13

---

### Purpose of the Windows 2000 Plug-In

The N1 Grid Service Provisioning System software provides enhanced capabilities in out-of-the-box support for Windows Internet Information Services (IIS), COM, and COM+ applications. You can capture an application from a Windows 2000 system, select precisely how this application should be configured, and deploy the application to stand-alone, managed, or clustered Windows 2000 environments.

For more information about Microsoft Windows IIS, see

<http://www.microsoft.com/resources/documentation/iis/6/all/proddocs/en-us/iis>

---

### What the Windows 2000 Plug-In Includes

The Windows 2000 Plug-In includes several Windows-specific component types, along with predefined components and resource files that enable you to easily capture, configure, and deploy Windows IIS, COM, and COM+ applications. The Windows 2000 Plug-In includes the following specific items:

3\_0\_1/resources/com/sun/windows/cmdInDir.bat  
3\_0\_1/resources/com/sun/windows/cscriptInDir.bat  
3\_1\_1/resources/com/sun/windows/IMSABOWrapper.dll  
fiji/components/com/sun/windows/compTypes/Batch File CT.xml  
fiji/components/com/sun/windows/compTypes/COM Component CT.xml  
fiji/components/com/sun/windows/compTypes/COMPlus CT.xml  
fiji/components/com/sun/windows/compTypes/DSN CT.xml  
fiji/components/com/sun/windows/compTypes/Global ISAPI Filter CT.xml  
fiji/components/com/sun/windows/compTypes/IIS Application CT.xml  
fiji/components/com/sun/windows/compTypes/IIS Global Settings CT.xml  
fiji/components/com/sun/windows/compTypes/IIS Site CT.xml  
fiji/components/com/sun/windows/compTypes/MSI CT.xml  
fiji/components/com/sun/windows/compTypes/Metabase Base CT.xml  
fiji/components/com/sun/windows/compTypes/Registry File CT.xml  
fiji/components/com/sun/windows/compTypes/Registry Key CT.xml  
fiji/components/com/sun/windows/compTypes/Website ISAPI Filter CT.xml  
fiji/components/com/sun/windows/compTypes/Windows Scripting Host CT.xml  
fiji/components/com/sun/windows/simpleComps/jacozoom/21100220.LIC.xml  
fiji/components/com/sun/windows/simpleComps/jacozoom/JacozoomContainer.xml  
fiji/components/com/sun/windows/simpleComps/jacozoom/izmcomjni.jar.xml  
fiji/components/com/sun/windows/simpleComps/jacozoom/izmjnicom.dll.xml  
fiji/components/com/sun/windows/simpleComps/jacozoom/winutils.jar.xml  
fiji/components/com/sun/windows/simpleComps/IMSABOWrapper.dll.xml  
fiji/components/com/sun/windows/simpleComps/cmdInDir.bat.xml  
fiji/components/com/sun/windows/simpleComps/cscriptInDir.bat.xml  
fiji/components/com/sun/windows/simpleComps/plugin-win32.jar.xml  
fiji/components/com/sun/windows/simpleComps/tmp\_cmdInDir.bat.xml  
fiji/components/com/sun/windows/simpleComps/tmp\_cscriptInDir.bat.xml  
fiji/components/com/sun/windows/sysServices/COMPlus SS.xml  
fiji/components/com/sun/windows/sysServices/DSN SS.xml  
fiji/components/com/sun/windows/sysServices/Metabase SS.xml  
fiji/components/com/sun/windows/sysServices/Registry Key SS.xml  
fiji/components/com/sun/windows/sysServices/Windows Services SS.xml  
fiji/resources/com/sun/windows/plugin-win32.jar  
fiji/resources/com/sun/windows/jacozoom/izmcomjni.jar  
fiji/resources/com/sun/windows/jacozoom/winutils.jar  
fiji/resources/com/sun/windows/jacozoom/21100220.LIC  
fiji/resources/com/sun/windows/jacozoom/izmjnicom.dll  
redmond/components/com/sun/windows/simpleComps/tmp\_cmdInDir.bat.xml  
redmond/components/com/sun/windows/simpleComps/tmp\_cscriptInDir.bat.xml  
redmond/resources/com/sun/windows/cmdInDir.bat  
redmond/resources/com/sun/windows/cscriptInDir.bat

---

# Requirements for Using the Windows 2000 Plug-In

## Requirements for Deployment Server

Any host on which you intend to deploy the Windows 2000 Plug-In must meet the following requirements:

Operating system	Appropriate version of the operating system
Disk space	Approximately 236 Mbytes of free storage space for the installed product and 170 Mbytes of temporary storage space required by the installer
RAM	256 Mbytes (minimum); 512 Mbytes (recommended)

## Requirements for Target Hosts

Any host on which you intend to deploy Windows 2000 files and applications must be running an appropriate version of the Windows 2000 operating system.



## Installing and Configuring the Windows 2000 Plug-In

---

This chapter provides information about installing and configuring the Windows 2000 plug-in. The chapter contains the following information:

- “Acquiring the Windows 2000 Plug-In” on page 15
- “Adding the Windows 2000 Plug-In to N1 Grid Service Provisioning System” on page 15
- “Customizing the Solution for Your Environment” on page 16

---

### Acquiring the Windows 2000 Plug-In

The Windows 2000 solution is packaged as a *plug-in* to the N1 Grid Service Provisioning System software. Plug-ins are packaged in Java™ Archive (JAR) files. The Windows 2000 plug-in files are available from the N1 Grid Service Provisioning System 5.0 Supplement CD or from the Sun Download Center.

---

### Adding the Windows 2000 Plug-In to N1 Grid Service Provisioning System

To make a given plug-in known to the N1 Grid Service Provisioning System product, you need to import the plug-in. To import a plug-in, follow these steps as explained in detail in Chapter 5, “Plug-In Administration,” in *N1 Grid Service Provisioning System 5.0 System Administration Guide*.

1. In the Administrative section of the main window, click Plug-ins.

2. In the Action column of the Plug-ins page, click Import.
3. Browse to the location where you downloaded the `com.sun.windows_1.1.jar` file.
4. Click the Continue to Import button.

When the import completes successfully, a plug-in details page appears that shows you the objects that the plug-in provides.

You can also import a plug-in archive file from the command line. Use the following command:

```
% cr-cli -cmd plg.p.add -path com.sun.windows_1.1.jar -u username -p password
```

---

## Customizing the Solution for Your Environment

You should grant the following permissions to the specified folders from the provisioning software interface:

- `/com/sun/windows` folder
  - Grant `allow on host set` permission to all users who will perform deployments that use the Windows 2000 Plug-In components on hosts that the users are expected to deploy to. Also grant this permission to all users who are expected to prepare Windows hosts on the hosts they are expected to prepare.
  - Grant `run component procedures` permission only as needed. This permission is recommended only in non-production systems, or for trusted admin users. This permission allows services of the `com.sun.windows#services` system service to be directly executed. These services include controls to start and stop services, execute batch and cscript files, and manage MSIs.
- `/com/sun/windows/jacozoom` folder
  - Grant `allow on host set` permission to all users who will be expected to prepare windows hosts on the hosts they are expected to prepare.



## Using the Windows 2000 Plug-In

---

This chapter explains how to capture and deploy Windows 2000 applications and files. The chapter includes the following information:

- “Capturing and Deploying Files and Applications” on page 17
- “Component Types” on page 20
- “Microsoft Windows System Services” on page 32
- “Extended Control Services” on page 32

---

### Capturing and Deploying Files and Applications

The Windows common tasks page enables you to view and create components for the following Windows 2000 entities:

- Windows applications, separated into three types:
  - COM+ applications
  - COM objects
  - Microsoft Windows installer files (.msi)
- Internet Information Server (IIS) applications and support files, separated into five types:
  - IIS applications
  - IIS web sites or virtual directory settings
  - IIS global settings
  - ISAPI global filter settings
  - ISAPI web site filter settings
- Other common Windows files, separated into five types:
  - Registry keys

- Registry files
- Data source names
- Batch files
- Scripting host scripts

## ▼ How to Capture a Windows 2000 File or Application

Before you can deploy a file or application across systems, you have to make the file or application known to the N1 Grid Service Provisioning System software and wrap it into a component.

---

**Note** – Although the general approach that you follow is the same for all component types, details differ depending on the actual types that you need to use.

---

- Steps**
1. **Select the Windows icon under the Common Tasks section of the left control panel.**
  2. **Click the Create New link next to the component type you need to create.**  
For example, to create a component that contains a batch file, follow these steps:
    - a. **Go to the Additional Windows Tasks section.**
    - b. **Click Create New next to Windows Batch File.**
  3. **On the Components Details Edit page, fill in the appropriate information.**  
For example, for a batch file, fill in this information:
    - a. **Type a name for the component**
    - b. **Choose the host from which to procure the batch file.**
    - c. **Navigate through the file system hierarchy on the host, and choose the batch file.**
  4. **Click Check in Selected Item.**

**More Information**

### Using the Command Line

You can also create a component through the command line. For more information about commands, see “cdb.rsrc: Managing Components” in *N1 Grid Service Provisioning System 5.0 Command-Line Interface Reference Manual*.

- For a simple file, such as a batch file, a command file, or a dynamic linked library file, use the following command: `cr_cli -cmd cdb.rsrc.ci -src filename -dst component-name -u username -p password`

where:

- *filename* is the name of the file that you want to check in.
- *component-name* is the name that you want to use for the component
- For files that you have to browse, such as COM+ applications, registry keys, or data source names, the command is more complex. If you have exported the file, you can use an “Exported File Browser” to check in the file. In the command, you have to specify the *sourcePath*. The *sourcePath* is the path that you would see when you browse for a file of this component type in the browser interface.

The following example checks in a data source name from a file *dsn.n1test*.

```
%> cr_cli -cmd cdb.rsrc.ci
      -src exported/dsn.n1test
      -dst dsn.n1test
      -type 'com.sun.windows#data source name'
      -platform 'NM:system#Windows 2000 Server'
      -pickerName 'Exported File Browser'
      -extraOpts sourcePath='System DSNn\\1test\\'
      -u **** -p ****
```

## ▼ How to Deploy a Windows 2000 File or Application

- Steps**
1. **Capture the component as described above.**
  2. **Go to the Components section of the N1 Grid Service Provisioning System application.**
  3. **Choose the component to deploy.**  
Depending on the type of component that you choose to deploy, you might have to define variables or to perform other steps. At a minimum, you perform the rest of the steps listed here.
  4. **Click Run in the Actions column next to the Default:Install procedure.**  
A plan details page appears for running the installation procedure.
  5. **Choose a target host.**  
The target host should be a host for which the component that you are deploying applies. For example, deploying a Microsoft Windows IIS application to a non-Windows host would not be appropriate.
  6. **Click Run Plan.**

---

# Component Types

To reflect the browser interface, the component types for this plug-in are grouped into three areas:

- Windows applications component types
- IIS component types
- Additional Windows component types

## Windows Applications Component Types

Windows application component types consist of three types:

- [“COM+ Application Component Type” on page 20](#)
- [“COM Component Component Type” on page 22](#)
- [“Microsoft Windows Installer File \(\\*.msi\) Component Type” on page 23](#)

### COM+ Application Component Type

A component of this type represents a COM+ application. COM+ applications are treated as a unit. The settings and content are installed as a group.

#### *Browsing for a COM+ Application*

When you browse for a COM+ application, the browser provides an alphabetical list of COM+ applications on the target system.

You can select a single COM+ application to be checked in. COM+ applications have no children and cannot be expanded..

#### *Exported/Internal File Format*

COM+ applications are exported into a Microsoft Windows Installer (MSI) file using the COM+ Admin SDK.

#### *Component Differences*

The COM+ Application is re-exported on the target system as an MSI file and compared against the MSI file used to install the application. The software will only indicate that there were differences, but will not indicate the details of the differences.

### *Installing a COM+ Application*

- If a COM+ application with the same name is already installed on the target system and running as a service, the existing application is stopped along with any running dependent services. The COM+ application will then be deleted from the COM+ catalog.
- The new COM+ application is installed using the COM+ Admin SDK.
- To start the COM+ application the user will have to use the `startApp` call step to manually start the COM+ application.

### *Uninstalling a COM+ Application*

The COM+ application is uninstalled using the COM+ Admin SDK.

### *Error Conditions*

**TABLE 3-1** COM+ Application Error Conditions

Action	Condition	Result
Install	COM+ Application already exists with the same name and either cannot be stopped or dependent services cannot be stopped.	Installation fails
Uninstall	MSI file used for installation is no longer available	Uninstallion fails
Install or Uninstall	Remote Agent does not have administrator privileges	Installation or Uninstallation fails

### *Extended Control Procedures*

**TABLE 3-2** COM+ Application Extended Control Procedures

Name	Parameters	Description
<code>startApp</code>	<i>appName</i> – Full name of the COM+ application.	Starts the COM+ application if it is run as a service
<code>stopApp</code>	<i>appName</i> – Fill name of the COM+ application to stop.	Stops the COM+ application and all dependent services
<code>stopRouter</code>	N/A	Stops the COM+ Routing services

**TABLE 3-2** COM+ Application Extended Control Procedures *(Continued)*

Name	Parameters	Description
startRouter	N/A	Starts the COM+ Routing services
installAsUser	<i>rsrcSrcPath</i> – Name of the COM+ application <i>rsrcInstallPath</i> – Path to the *.msi file representing the application <i>userID</i> – User who is going to run the application <i>password</i> – Password of the user	Enables installation of a COM+ application that runs as a particular user

## COM Component Component Type

This component type represents a COM file.

### *Browsing*

To choose a file for this component type, select a \*.ocx or \*.dll file from the file browser.

### *Component Differences*

The COM component is compared as a binary file against the file used during installation. The software indicates when there are differences, but does not provide details about the differences.

### *Installing a COM Component*

The Regsvr32 utility is called to register the COM components in the DLL using the following command line:

```
regsvr32.exe /s file path
```

### *Uninstalling a COM Component*

The Regsvr32 utility is called to unregister the COM components in the DLL using the following command line:

```
regsvr32.exe /s /u <file path>
```

After the DLL is unregistered, it is removed from the target system.

### *Error Conditions*

**TABLE 3-3** COM Component Error Conditions

<b>Action</b>	<b>Condition</b>	<b>Result</b>
Install	The supplied .dll or .ocx file does not contain COM components	Installation fails
Uninstall	The supplied .dll or .ocx file does not contain COM components	Uninstallation fails

## Microsoft Windows Installer File (\*.msi) Component Type

A component of this component type includes a silent Microsoft Installer (MSI) file.

### *Browsing*

To choose a file for this component type, select a \*.msi file from the file browser.

### *Installing an MSI File*

The Microsoft Windows installer service is called on the MSI file to import it into the target system with the following command:

```
msiexec /qn /i file path
```

### *Uninstalling an MSI File*

The Microsoft Windows installer service uninstall is called on the MSI file used during installation to uninstall the package using the following command:

```
msiexec /qn /x file path
```

After msiexec finishes, the MSI file is removed.

## Error Conditions

**TABLE 3-4** Microsoft Windows Installer File Error Conditions

Action	Condition	Result
Install	The supplied *.msi file is not a proper Microsoft Windows installer file.	Installation fails.
Install	The agent does not have proper permissions to run installations	Installation fails.
Uninstall	The package has already been uninstalled	Uninstall fails

## IIS Component Types

IIS component types share some common information and consist of several component types:

- [“Common IIS Component Type Formats, Errors, and Functions” on page 24](#)
- [“IIS Global Filter Settings Component Type” on page 26](#)
- [“IIS Web Site Filter Settings Component Type” on page 26](#)
- [“IIS Global Settings Component Type” on page 27](#)
- [“IIS Website \(IIS Virtual Directory\) Component Type” on page 27](#)
- [“IIS Website Filter Component Type” on page 28](#)

## Common IIS Component Type Formats, Errors, and Functions

The IIS component types share a common implementation. All IIS component types enable you to export, install, and delete data stored in the IIS metabase. To enable the common behaviors, these component types share a set of functions, formats, and errors.

### *Common IIS Browsing*

When you browse for an IIS component, the browser provides a list of web sites and virtual directories on the target system. These items appear in the order that they occur in the metabase. This corresponds to the order in which the items appear in the IIS Control panel. This list is *not* alphabetical.

You can select a single web site or virtual directory. Selecting a web site is considered recursive. Double-clicking a web site displays a list of the virtual directories in the web site.



### *Exported/Internal File Format*

All IIS types use an XML format to store their section of the metabase. The present XML format does not support metabase properties of type NTACL (such as AdminACL). Any properties of this type that are encountered while reading from or writing to the metabase are ignored. Also, properties of type IPsec (such as IPSecurity) are written out as serialized objects. Those objects are not human-readable during either direct examination, or as difference results.

### *Differences*

During a snapshot, the current state of the metabase is exported into an XML file. During an M-I difference, the metabase is re-exported and compared against the original XML file. The standard XML differentiator is used to generate differences between these files.

### *Error Conditions*

**TABLE 3-5** IIS Types Common Error Conditions

Action	Condition	Result
Install/Export	IIS Does not exist or is the improper version	Install/Export fails
Install/Uninstall	Remote Agent does not have administrator privileges	Install/Uninstall fails

### *IIS Web Site or Virtual Directory Settings*

Represents the settings for an IIS web site or virtual directory. Please note that this component type only contains the settings for a web site or virtual directory. The content on the web site must be checked in as a separate resource.

### *Installing an IIS Web Site or Virtual Directory*

Installation occurs by reading the XML file and importing the file into the target system metabase. If a web site with the same name exists, the old web site is overwritten. If multiple web sites have the same name on the system, the first matching web site is removed and overwritten.

Special cases include untyped keys or nodes, and SSL certificates, which are not deployed. The relevant settings for SSL certificates in IIS (SSLCertHash and SSLStoreName) are preserved during a deployment if they exist on the target system. If relevant SSL settings do not exist, settings are not added.

To bring up a secure site after the site is deployed or redeployed, you must restart IIS.

### *Uninstalling an IIS Web Site or Virtual Directory*

The entire web site is removed on the target system. All virtual directories in the web site are removed regardless of whether they were installed by the provisioning software. If the settings are just for a virtual directory, only that directory is removed, not its containing site. The matching for uninstall is done by name, so the first web site with the same name found on the system will be uninstalled. Once this process is complete, the XML file used during installation is removed.

## IIS Global Filter Settings Component Type

This component type is used to represent global IIS filter settings. This component type only contains the settings for an IIS Global Filter. You must install separately the actual DLL that implements the filter.

### *Browsing for an IIS Global Filter*

When you browse for an IIS global filter settings component, the browser provides a list of global filters on the target system. These filters appear in the order that they occur in the metabase. This corresponds to the order in which the items appear in the IIS Control panel. This list is *not* alphabetical.

You can select a single filter. Filters cannot be expanded.

### *Installing an IIS Global Filter*

Installation occurs by reading the XML file and importing the file into the target system metabase. If a filter setting with the same name exists on the target machine, the existing filter setting is overwritten.

### *Uninstalling an IIS Global Filter*

The filter settings are removed on the target system. Once the settings are removed, the XML file used during installation is removed.

## IIS Web Site Filter Settings Component Type

This component type contains the settings for a web site filter. You must install separately the actual DLL file that implements the filter.

### *Browsing for Web Site Filter Settings*

When you browse for an IIS web site filter settings component, the browser provides a list of web site filters on the target system. These filters appear in the order that they occur in the metabase. This corresponds to the order in which the items appear in the IIS Control panel. This list is *not* alphabetical.

To see a list of filters for a web site, you must expand the web site. You can select all web site filters or an individual filter to be checked in.

### *Installing Web Site Filter Settings*

Installation occurs by reading the XML file and importing the file into the target system metabase. If a filter setting with the same name exists on the target machine, the existing filter setting is overwritten.

### *Uninstalling Web Site Filter Settings*

The filter settings are removed on the target system. Once the settings are removed, the XML file used during installation is removed.

## IIS Global Settings Component Type

This component type is used to represent global IIS settings.

### *Browsing for Global Settings*

When you browse for an IIS global settings component, the browser provides a list of settings on the target system. These settings appear in the order that they occur in the metabase.

You can select a single setting to be checked in. Settings have no children and cannot be expanded.

### *Installing Global Settings*

Installation occurs by reading the XML file and importing the file into the target system metabase. The setting on the target system is overwritten if the setting exists.

### *Uninstalling Global Settings*

Global settings cannot be uninstalled. Uninstalls will have no affect on the target system except to remove the XML file used during install.

## IIS Website (IIS Virtual Directory) Component Type

The IIS Website (IIS Virtual Directory) component type is a composite component type that includes four components:

- A component that contains IIS web site or virtual directory settings

- A component that identifies an IIS virtual directory set
- A component that defines an IIS web site filter set
- A component that contains a directory

## IIS Website Filter Component Type

The IIS Website Filter component type is a composite type that includes two components:

- A component that contains IIS web site filter settings
- A component that contains a COM object

## Additional Windows Component Types

The Windows 2000 Plug-In also provides several additional component types:

- “Registry Keys Component Type” on page 28
- “Registry File Component Type” on page 29
- “Data Source Name Component Type” on page 30
- “Microsoft Windows Batch File Component Type” on page 31
- “Microsoft Windows Scripting Host Script Component Type” on page 31

## Registry Keys Component Type

This component type is used to represent registry keys and their associated values.

### *Browsing*

When you browse for a registry key component, the browser provides a list of registry keys from the main registry roots on the target system. The main registry roots include HKEY\_LOCAL\_MACHINE, HKEY\_CLASSES\_ROOT, HKEY\_CURRENT\_USER, HKEY\_USERS, and HKEY\_CURRENT\_CONFIG. These items appear in the order that they are presented in the registry.

You can select an individual key for check in. Selecting a key will check in that key and all of its children. Double clicking on a key will recursively check down the registry until a value is found. The name of the value is displayed but not its contents. Values can be individually exported.

### *Exported/Internal File Format*

Registry keys are exported into an XML file.

### *Differences Between Registry Key Versions*

During a snapshot, the current state of registry key (and its children) is exported into an XML file. During a difference check, the registry key is re-exported and compared against the original XML file. The standard XML difference comparator is used to generate differences between these files.

### *Installing a Registry Key*

The XML file that represents the registry is read and imported into the target system using an `execJava` step. Any keys for which values already exist in the target system are overwritten.

### *Uninstalling a Registry Key*

The `execJava` implementation takes the root of the exported key, and deletes all keys and values beneath it. If the root is a value, it will be deleted.

### *Error Conditions*

**TABLE 3-6** Registry Keys Error Conditions

Action	Condition	Result
Install or Uninstall	Remote agent does not have administrator privileges	Installation or uninstallation fails

## Registry File Component Type

A component of this component type contains a registry file (\*.reg). Registry files are text files that specify the key and values to add or remove from the registry metabase.

---

**Note** – The provisioning software does not compare versions of a registry file for differences. To be able to view differences between registry files, use the `Registry Keys` component type instead.

---

### *Browsing for a Registry File*

To choose a file for this component type, select a \*.reg file from the file browser.

### *Installing a Registry File*

`Regedit /s file path` is called on the \*.reg file to write its changes to the registry.

## *Uninstalling a Registry File*

During uninstallation, only the \*.reg file used during installation is removed. The registry keys inside the registry file are unaffected. Use the Registry Keys type to enable registry uninstallation.

## *Error Conditions*

**TABLE 3-7** Registry File Error Conditions

Action	Condition	Result
Install	The supplied *.reg file is not in the proper format for regedit.	Installation fails
Install	The agent does not have proper permissions to write into the registry sections designated by the *.reg file.	Installation fails

## Data Source Name Component Type

A component of this component type represents ODBC settings for connecting to a database. You can modify these settings through the Data Source Administrator control panel on the system. The actual settings are stored in specific places in the registry. As a result, the Data Source Name (DSN) resource type is built on top of the Registry Keys resource type. The DSN installation, export, and uninstallation functions directly use the facilities provided by the Registry Key resource handler. The DSN browser mimics the Registry browser to provide an experience closer to the "Data Source Administrator" control panel.

### *Browsing for a DSN File*

When you browse for a DSN component, the browser provides a list of DSN files from the main registry roots on the target system. The DSN roots include User and System. DSN files are displayed in alphabetical order.

You can select an individual DSN file for check in. Double-click on the System and User roots to list the contained DSN entries.

### *Exported/Internal File Format*

On export, the browser exports the key that contains all the DSN settings, as well as the value of the same name in the "ODBC Data Sources" key at the same level in the registry hierarchy.

### *Uninstalling a DSN File*

The DSN uninstallation is based on the registry uninstallation, with the caveat that the path being deleted is the key that contains the DSN settings but not the key that the DSN user interface uses to display the available DSN settings. Special logic exists to delete this key as well. The semantics of this process differ slightly from the registry uninstallation semantics, although both processes use the same executor.

### *Error Conditions*

The DSN system component directly calls the Install method of the registry system component. See [“Registry Keys Component Type” on page 28](#) and [“Registry Keys Component Type” on page 28](#) for further information on implementation and possible errors.

## Microsoft Windows Batch File Component Type

A component of this component type contains a Windows batch \*.bat or command \*.cmd file.

### *Browsing*

When you create a component for this component type, you can request that the file browser show only that files that have \*.bat or \*.cmd extensions.

### *Error Conditions*

**TABLE 3-8** Microsoft Windows Batch File Error Conditions

Action	Condition	Result
Install	The supplied batch file is not a valid batch file or contains errors.	Installation fails

## Microsoft Windows Scripting Host Script Component Type

A component of this component type contains Microsoft Windows scripting host (WSH) scripts. These files are text files that are created by either vbscript (\*.vbs) or jscript (\*.js). These files may also be contained in an XML project file (\*.wsf).

### *Browsing for a Scripting Host Script*

When you create a component for this component type, you can request that the file browser show only that files that have \*.js, \*.vbs, or \*.wsf extensions.

## *Uninstalling a Scripting Host Script*

When you uninstall a component of this component type, the script file is removed from the target host. However, any actions that the script performed are not undone.

## *Error Conditions*

**TABLE 3-9** Microsoft Windows Scripting Host Script Error Conditions

Action	Condition	Result
Install	The supplied file is not a valid *.wsf file or contains errors.	Installation fails

---

# Microsoft Windows System Services

The Microsoft Windows System Component provides utilities for interacting with different windows systems, beyond those related to any specific component types. The plug-in includes five Windows-specific system services:

- com.sun.windows#COM+ application services
- com.sun.windows#data source name services
- com.sun.windows#metabase services
- com.sun.windows#registry key services
- com.sun.windows#services

---

# Extended Control Services

Extended control services are procedures that perform a software operation related to a resource or component. These services include services for starting and stop the Microsoft IIS Web Server and for starting and stopping specific Microsoft Windows services. The provisioning software includes two sets of extended control services that are not specific to individual resource types:

- General purpose control services, which can be called by any component or plan
- Resource-type-specific control services, which are related to a specific type of resource and that can be called only by components or plans that include resources with those resource types

General purpose control services include the ability to start or stop Microsoft Windows services. These control services can be called by any component or plan.



An example of a resource-type-specific control service is the `startRouter` procedure included with the COM+ Application resource type. This procedure starts COM+ routing services. It works only with resources that are of the COM+ Application resource type.

## Extended Control Services for IIS

The following table lists the procedures for controlling IIS.

**TABLE 3-10** Built-in Procedures for Controlling IIS

Name	Parameters	Description
<code>stopIIS</code>	None	Shuts down the IIS process
<code>startIIS</code>	None	Restarts IIS if it is not running

## Extended Control Services for Microsoft Windows Services

A component or plan may include steps to start or stop specific Microsoft Windows Services.

The following table describes procedures for controlling Microsoft Windows Services.

**TABLE 3-11** Built-in Procedures for Controlling Microsoft Windows Services

Name	Parameters	Description
<code>stopService</code>	<i>serviceName</i> – Name of the service to stop.	Shuts down the service
<code>startService</code>	<i>serviceName</i> – Name of the service to start.	Starts the Microsoft Windows service



# Index

---

## C

capturing an application, 18-19  
configuring the plug-in, 16

## D

deployment server requirements, 13

## I

importing the plug-in, 15-16

## J

JAR file, 15  
Java archive, *See* JAR file

## P

plug-in files  
importing, 15-16  
location of, 15

## T

target host requirements, 13

## W

windows plug-in  
configuration, 16  
Windows plug-in  
deployment server requirements, 13  
windows plug-in  
importing, 15-16  
Windows plug-in  
target host requirements, 13

