## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Welcome</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>About This Document</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Terms and Conventions</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Feedback</td>
<td>9</td>
</tr>
<tr>
<td>Section 1</td>
<td>DesktopNow Introduction</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>About DesktopNow Products</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Console</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Configuration File</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Agent</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Application Manager</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>AM Web Service</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Environment Manager</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Environment Manager Administrative Tools</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Personalization Operations</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Performance Manager</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Management Center</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Management Center Components</td>
<td>20</td>
</tr>
<tr>
<td>Section 2</td>
<td>Prerequisites</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Supported Languages</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Supported Operating Systems and Technologies</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>System Requirements</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Required Components</td>
<td>25</td>
</tr>
</tbody>
</table>
## Section 7: Patch Installation

### Patching

- Install and Uninstall a Patch Using the Management Center
  - Upload an MSP
  - Deploy an MSP to an endpoint
  - Uninstall an MSP

- Install and Uninstall a Patch Using the Command Line
  - Command Line Install
  - Command Line Uninstall

- Roll Back Service Packs
  - Rolling Back Service Packs Using Windows Control Panel
  - Rolling Back Service Packs Using Management Center

## Section 8: Product Upgrades

### Preparing to Upgrade

### Upgrade with the DesktopNow Installer

### Upgrade Application Manager

- Upgrades and Process Rules
- Upgrades and Group Management
- URL Redirection and Custom Rules

### Upgrade Environment Manager

- Endpoint Configuration Merging
- Conversion Rules
- Upgrading The Logon Trigger
- Upgrading Certificates

- Personalization Operations Bulk Operations User Selection

### Upgrade Configurations

### Upgrade Servers

## Section 9: Uninstall

### Uninstallation

## Appendices

### Appendix A: Multi Instance Command Line Installer
The Command Line Installer  117
Using InstallerCmd.exe  117
InstallerCmd.exe Examples  118

Appendix B  Management Center MSI Custom Actions Usage  119
MSI Custom Actions  120
Management Console Custom Actions  120
Management Server Custom Actions  127
Client Communications Agent Custom Actions  134

Appendix C  Management Center Third Party Public Symbols Usage  138

Glossary  140
Welcome

In this Section:
- About This Document on page 9
- Terms and Conventions on page 9
- Feedback on page 9
About This Document

This guide takes you through the steps required to install, upgrade and uninstall the AppSense DesktopNow products: Application Manager, Environment Manager and Performance Manager.

The guide also provides a basic orientation to the product consoles.

Terms and Conventions

The following tables show the textual and formatting conventions used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Highlights items you can select in Windows and the product interface, including nodes, menus items, dialogs and features.</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>Used for scripting samples and code strings.</td>
</tr>
<tr>
<td><strong>Italic</strong></td>
<td>Highlights values you can enter in console text boxes and titles for other guides and Helps in the documentation set.</td>
</tr>
<tr>
<td><strong>Green + underlined</strong></td>
<td>Indicates a glossary link.</td>
</tr>
<tr>
<td><strong>&gt;</strong></td>
<td>Indicates the path of a menu option. For example, &quot;Select File &gt; Open&quot; means &quot;click the File menu, and then click Open.&quot;</td>
</tr>
</tbody>
</table>

Feedback

The AppSense Documentation team aim to provide accurate and high quality documentation to assist you in the installation, configuration and ongoing operation of AppSense products.

Please provide any feedback to: documentation.feedback@appsense.com
DesktopNow Introduction

In this Section:

- About DesktopNow Products on page 11
- Application Manager on page 17
- Environment Manager on page 18
- Performance Manager on page 19
- Management Center on page 20
About DesktopNow Products

AppSense DesktopNow product suite consists of: Application Manager, Environment Manager, and Performance Manager.

AppSense Management Center is the framework that enables the AppSense products, Application Manager, Environment Manager, and Performance Manager, to be used across an entire enterprise.
The DesktopNow console launches when the link is selected in the Start > All Programs > AppSense menu.
File Menu
The File menu provides options for managing configurations.

File Menu Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Creates a new default configuration that is locked for editing.</td>
</tr>
</tbody>
</table>
| Open   | Opens an existing A*MP configuration file for editing from one of the following locations:  
- Live configuration on this computer.  
- Configuration from the Management Center.  
- Configuration file on a local or network drive: DesktopNow Product Package Files format (a*mp).  
- A configuration from the System Center Configuration Manager.  
**Note** A live configuration is located on a computer which has an DesktopNow Agent installed and running. |
| Save   | Saves the configuration in one of the following states:  
- Save and continue editing - save the configuration and keep it locked and open for editing. You will not be able to deploy the configuration while it is locked.  
- Save and unlock - save the configuration and unlock it ready for deployment. The current configuration closes and a new default configuration opens.  
- Unlock without saving - unlock the configuration without saving changes. The current configuration closes and a new default configuration opens. |
| Save As| Saves the configuration with a new name to one of the following locations:  
- Live configuration on this computer  
- Configuration in the Management Center  
- Configuration in System Center Configuration Manager  
- Configuration in Group Policy  
- Configuration file on a local or network drive: DesktopNow Product Package Files format (a*mp).  
**Note** A live configuration is located on a computer that has a DesktopNow Agent installed and running. |
| Import & Export | Imports a configuration from MSI format, usually legacy configurations that have been exported and saved from legacy consoles.  
- Exports a configuration to MSI format. |
| Exit   | Closes the console.  
You are prompted to save any changes you have made to the current configuration. |

* - replace with first letter of product i.e. Application Manager = A, Environment Manager = E, Performance Manager = P

Quick Access Toolbar
The Quick Access toolbar provides quick functionality for managing the configuration.
### Quick Access Toolbar Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Save](image) | Save
Saves changes to the configuration. The configuration will remain locked if opened from the AppSense Management Center. |
| ![Save and unlock](image) | Save and unlock
Saves changes and unlocks the configuration. These changes can now be deployed from the AppSense Management Center. |
| ![Save As](image) | Save As                                                                                                                                   |
| ![Undo](image) | Undo
Clears the action history. Up to 20 previous actions are listed. Select the point at which you want to clear the actions. The action selected and all proceeding actions are undone. |
| ![Redo](image) | Redo
Re-applies the cleared action history. Up to 20 cleared actions are listed. Select the point at which you want to redo the actions. The action selected and all subsequent actions are redone. |
| ![Back](image) | Back
Navigates back through the views visited in this session. |
| ![Forward](image) | Forward
Navigate forward through the views visited this session. |
| ![New](image) | New
Opens a new, empty default configuration which is locked for editing. If you already have a configuration open, you will be prompted to save it before you open a new one. |
| ![Configuration](image) | Configuration
Open an existing configuration from the Management Center. |
| ![Expand All](image) | Expand All
Expand all nodes. |
| ![Collapse All](image) | Collapse All
Collapses all nodes. |
Managing the Quick Access Toolbar
You can configure the Quick Access toolbar to display the commands you use the most and to change its position in the console:

- Right-click on a ribbon button or file menu option and select Add to Quick Access Toolbar to add it to the Quick Access Toolbar.
- Right-click on a toolbar item and select Remove From Quick Access Toolbar to remove it.
- Right click on a ribbon or the toolbar and select Show Quick Access Toolbar Below the Ribbon to display the toolbar below the ribbon.

Ribbons
Ribbons group together buttons according to their functionality or area of the console. For example, the Home ribbon includes all common tasks such as About, Cut, Paste and Copy, Help, AppSense website, and Support links.

Split ribbon buttons contain multiple options and are indicated by an arrow just below the button. Click the arrow to display and select the list of options, or simply click the button for the default action.

Help
The User Assistance Portal provides links to all AppSense product Help systems, Product Guides and Supporting product documentation:

www.appsense.com/help

You can launch the Help from within the product consoles in the following ways:

- Help ribbon button - displays the Help landing page and the Table of Contents.
- Help icon top right corner of the console or dialog - displays the context sensitive help related to the current view.
- Press F1 for context sensitive help related to the current view.

Navigation Button
The Navigation Button changes the console and navigation tree view.

Navigation Tree
The Navigation Tree has nodes and sub nodes that contain settings to configure and manage the configuration. It is context-sensitive depending on which navigation button is selected.

Work Area
The Work area provides the main area for managing the settings of the configuration and product. The contents of the work area vary according to the selected nodes in the navigation tree and the selected navigation buttons.

Actions Panel
The Management Console has an Actions panel that displays in the right-hand column and shows available controls for the current view.
Additional Console Features

- Shortcut Menu — right-click shortcuts are available in the navigation tree and some areas of the console.
- Drag and Drop — this feature is available in some nodes of the navigation tree.
- Cut/Copy/Paste — these actions can be performed using the buttons in the Edit ribbon in Application Manager and Environment Manager, and the Home ribbon in Performance Manager, shortcut menu options and also using keyboard shortcuts.

Configuration File

Configuration files contain the settings created using the product console.

Configurations are stored locally in the All Users profile and are protected by NTFS security. In Evaluation mode, configuration changes are saved in the custom .a*mp format (AppSense [product] Manager Package) and read by the agent.

In Advanced mode, configurations are stored in the AppSense Management Center database, and deployed using the AppSense Management console.

Configurations can be exported and imported to and from MSI file format using the product consoles. This is useful for creating templates or distributing configurations using third-party deployment systems.

After creating or modifying a configuration, you must save the configuration with the latest settings to ensure that they are implemented.

Agent

DesktopNow products are installed and run on endpoints using a lightweight agent. The agent is deployed to managed computers to implement the configuration rules. In Evaluation mode, the agent is installed directly onto the local computer. In Advanced mode, configurations are stored centrally and deployed remotely across a network to multiple controlled computers using the AppSense Management Center.

Agents are constructed as Windows Installer MSI packages, which allows them to be distributed using any third-party deployment system that supports the MSI format.

Note
For more information about deploying AppSense DesktopNow products, see the AppSense Management Center Help.
Application Manager

Use Application Manager to control which applications a user receives on their physical or virtual desktop.

Application Manager provides protective measures, such as automatically blocking the execution of all authorized applications, eliminating the threat of a user introducing - either intentionally or unintentionally - an executable file to a network.

The product gives you granular control so that you can decide at user level who has the authority to run specific applications.

AM Web Service

The AM Web Service is installed on any selected machine as part of the Application Manager installation. It is a lightweight component that does not require typical server tools such as Internet Information Services (IIS) or SQL Server. In Evaluation mode, the service is installed on any selected machine. To install the Service as part of the Advanced mode, the Application Privilege Discovery option must be selected.

Note
For more information about Application Manager and Application Manager Web Services, see the AppSense Application Manager Help.
Environment Manager

AppSense Environment Manager is a user virtualization solution that ensures users always receive a consistent, predictable, and personalized working experience.

User virtualization represents a fundamental change in the way the corporate desktop is constructed, delivered, and managed.

Environment Manager enables standard desktop images to be used to deliver fully configured and personalized desktops to all employees. The user component of a desktop (user personality) is decoupled from the operating system and applications, managed independently, and applied into a desktop as required. This is achieved without scripting, group policies, or use of roaming user profiles - regardless of how the desktop is delivered.

Environment Manager provides a more efficient alternative to roaming profiles, reducing the potential for profile corruption and providing users with a consistent and seamless working experience.

Because Environment Manager applies user data on demand, you can combine delivery methods, migrate users between platforms and operating systems, and update corporate desktops without impacting user experience.

Environment Manager uses a combination of Policy Configuration and User Personalization to enable comprehensive user virtualization.

- **Policy Configuration** - Use corporate policy to set up a corporate desktop environment and specify what users can access, how they access it and what they can do with it.

- **User Personalization** - Constitutes anything a user is able to customize on their endpoint. This includes the desktop look and feel, application menus and buttons, language settings, and screen resolution.

AppSense Environment Manager enables users to have a single personality that is accessible from any location and any device. The settings save on disconnection, enabling the user to work offline. Any changes made to their personality while they are offline are synchronized with the corporate network when they reconnect.

The Environment Manager user interface provides the ability to create and modify both Policy and Personalization configurations. Both of these are created and deployed to endpoints in different ways. Policy configuration settings are stored in the AEMP configuration file, which must be deployed to endpoints before it can take effect. For User Personalization, the user interface maintains a live connection to the database. Changes are immediately saved to the database and take effect at each endpoint the next time it performs a configuration poll.

Note
For more information about Environment Manager see the AppSense Environment Manager Help.
Environment Manager Administrative Tools

Environment Manager is packaged with the following standalone tools, which assist administrators to create configurations and work with the Personalization Database.

- Client Debug Setup
- Personalization File Utility
- Personalization Migrate Utility
- Personalization Registry Utility
- Log Viewer
- File Based Registry Explorer

Note
For further information, see the AppSense Environment Manager Administrative Tools Guide available from the User Assistance Portal.

Personalization Operations

AppSense Personalization Operations is an AppSense Environment Manager utility that provides management of Personalization data via a web console. Depending on their role, users can manage backups and current settings for either single users or multiple users at a time. They can also search for and delete audit logs, and view the migration status of Personalization Groups.

Note
For further information, see the Personalization Operations Help.

Performance Manager

Use Performance Manager to implement rules to manage precisely the allocation and distribution of CPU, memory, and disk resources for applications and users on your system. Performance Manager includes automated application memory optimization to reduce page file use and CPU thread throttling to control demand on resources and ensure the efficient and smooth running of the system.

Performance Manager provides a fine level of granular management to allocate resources based on the state of a session, applications, or the desktop.

Note
For further information, see the Performance Manager Help.
Management Center

AppSense Management Center is a scalable multi-tier system that enables the central management and secure deployment of configuration information to thousands of endpoint devices and user environments. The Management Center incorporates comprehensive auditing and reporting, and provides failover support for server resiliency.

The Management Center, comprises of the Management Server, Database (Microsoft SQL Server), Management Console, and the Deployment Agent (CCA) which must be installed on each managed endpoint.

The Deployment Agent (CCA) uploads event data from managed endpoints via the Management Server to the database. Product configurations are created in the product consoles and stored in the Management Center database, from where they can be downloaded along with product agents and software updates by the Deployment Agent (CCA) for installation on managed endpoints.

Management Center Components

The Management Center includes the following components:

**Management Console**

The Management Console provides an interface to the Management Server and the other components of the Management Center, allowing you to control deployment groups, users, event data and alerts, configurations and packages, managed endpoints, and reports.

**Management Server**

The Management Server manages communications (using Microsoft Internet Information Services - IIS) with a Microsoft SQL Server database for data access and storage, providing security control, communications for managing network discovery services and software deployment to managed endpoints, resource management, and auditing.

**Database**

The Management Center must have access to a Microsoft SQL Server database for the storage and retrieval of AppSense software agents, configuration packages, license packages, and event and alert data.

The Management Server and can be installed locally on the Management Center host computer or on a remote computer.
Deployment Agent (CCA) on Managed Computers

The Deployment Agent (CCA) is installed on managed endpoints to manage communications between the product agents and the AppSense Management Center.

You can deploy the Deployment Agent (CCA) as follows:

- Use the Install CCA functionality in the Management Console.
- On each endpoint download the agent from the Management Server website and install it.
- Use a third-party deployment mechanism.

Note
For further information, see the AppSense Management Center Help.
Prerequisites

In this Section:

- Supported Languages on page 23
- Supported Operating Systems and Technologies on page 23
- System Requirements on page 23
- Required Components on page 25
Supported Languages

- English
- German
- French

Supported Operating Systems and Technologies

The Supported Operating Systems and Technologies are detailed in the Maintained Platforms Matrices available on AppSense Support Portal.

In the AppSense Support Portal, select View Maintained Platform Matrices under Our Software. Select your required matrix from the list of results. You can use the filter boxes to further refine the list.

System Requirements

The table below contains the minimum and recommended hardware requirements for running DesktopNow.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DesktopNow Agents</td>
<td>See Microsoft system requirements for the specific Windows version.</td>
</tr>
<tr>
<td>DesktopNow Consoles</td>
<td></td>
</tr>
<tr>
<td>Processor</td>
<td><strong>Minimum speed:</strong> 1 GHz (x86) or 1.4 GHz (x64)</td>
</tr>
<tr>
<td></td>
<td><strong>Recommended speed:</strong> 2 GHz or faster</td>
</tr>
<tr>
<td></td>
<td><strong>Minimum # of CPUs:</strong> 1</td>
</tr>
<tr>
<td></td>
<td><strong>Recommended # CPUs:</strong> 2 or greater</td>
</tr>
<tr>
<td></td>
<td>Refer to Windows editions documentation on support for more than 4 CPUs</td>
</tr>
<tr>
<td>Memory</td>
<td><strong>Minimum:</strong> 2 GB RAM</td>
</tr>
<tr>
<td></td>
<td><strong>Recommended:</strong> 4 GB RAM or greater</td>
</tr>
<tr>
<td></td>
<td>Refer to Windows editions documentation on support for more than 4 GB RAM</td>
</tr>
<tr>
<td></td>
<td>(x86) and 32 GB RAM (x64)</td>
</tr>
<tr>
<td>Available Disk Space</td>
<td><strong>Minimum:</strong> 1 GB</td>
</tr>
<tr>
<td>Recommended Resolution</td>
<td>1024 x 768 pixels</td>
</tr>
</tbody>
</table>
## System Requirements

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Personalization Server</strong></td>
<td></td>
</tr>
<tr>
<td>Processor</td>
<td>Minimum speed: 1.4 GHz (x64)</td>
</tr>
<tr>
<td></td>
<td>Recommended speed: 2 GHz or faster</td>
</tr>
<tr>
<td></td>
<td>Minimum # of CPUs: 1</td>
</tr>
<tr>
<td></td>
<td>Recommended # CPUs: 2 or greater</td>
</tr>
<tr>
<td></td>
<td>Refer to Windows editions documentation on support for more than 4 CPUs</td>
</tr>
<tr>
<td>Memory</td>
<td>Minimum: 2 GB RAM</td>
</tr>
<tr>
<td></td>
<td>Recommended: 4 GB RAM or greater</td>
</tr>
<tr>
<td></td>
<td>Refer to Windows editions documentation on support for more than 32 GB RAM (x64)</td>
</tr>
<tr>
<td>Available Disk Space</td>
<td>Minimum: 10 GB</td>
</tr>
<tr>
<td><strong>Personalization Database Server</strong></td>
<td></td>
</tr>
<tr>
<td>As the system requirements for the specific SQL Server version, plus 2 GB of available disk space for the initial installation.</td>
<td></td>
</tr>
<tr>
<td><strong>Application Manager</strong></td>
<td></td>
</tr>
<tr>
<td>Web Services</td>
<td></td>
</tr>
<tr>
<td>Processor</td>
<td>Minimum speed: 1 GHz (x86 or x64)</td>
</tr>
<tr>
<td>Memory</td>
<td>Minimum: 1 GB RAM (x86) or 2 GB RAM (x64)</td>
</tr>
<tr>
<td>Available Disk Space</td>
<td>Minimum: 10 GB</td>
</tr>
<tr>
<td>Device</td>
<td>DirectX9 graphics device with WDDM 1.0 or higher driver</td>
</tr>
<tr>
<td><strong>Management Server</strong></td>
<td></td>
</tr>
<tr>
<td>Processor</td>
<td>Minimum speed: 1.4 GHz (x64)</td>
</tr>
<tr>
<td></td>
<td>Recommended speed: 2 GHz or faster</td>
</tr>
<tr>
<td></td>
<td>Minimum # of CPUs: 1</td>
</tr>
<tr>
<td></td>
<td>Recommended # CPUs: 2 or greater</td>
</tr>
<tr>
<td></td>
<td>Refer to Windows editions documentation on support for more than 4 CPUs</td>
</tr>
<tr>
<td>Memory</td>
<td>Minimum: 2 GB RAM</td>
</tr>
<tr>
<td></td>
<td>Recommended: 4 GB RAM or greater</td>
</tr>
<tr>
<td></td>
<td>Refer to Windows editions documentation on support for more than 32 GB RAM (x64)</td>
</tr>
<tr>
<td>Available Disk Space</td>
<td>Minimum: 10 GB</td>
</tr>
</tbody>
</table>
Prerequisites

Required Components

The following components are installed as part of the AppSense DesktopNow Installer:

- Microsoft Windows Installer 5.0
- Web Server Internet Information Services (IIS)
  - Common HTTP Features
    - Default Document
    - Directory Browsing
    - HTTP Errors
    - Static Content
    - HTTP Redirection
  - Health and Diagnostics
    - HTTP Logging
    - Logging Tools
    - Request Monitor
    - Tracing
  - Performance
    - Static Content Compression
    - Dynamic Content Compression

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| Management Console| **Processor**
  - Minimum speed: 1 GHz (x86) or 1.4 GHz (x64)
  - Recommended speed: 2 GHz or faster
  - Minimum # of CPUs: 1
  - Recommended # CPUs: 2 or greater
  - Refer to Windows editions documentation on support for more than 4 CPUs |
|                   | **Memory**
  - Minimum: 2 GB RAM
  - Recommended: 4 GB RAM or greater
  - Refer to Windows editions documentation on support for more than 4 GB RAM (x86) and 32 GB RAM (x64) |
|                   | **Available Disk Space**
  - Minimum: 1 GB |
| Database Server   | As the system requirements for the specific SQL Server version, 2012 SP3 or above, 2014 SP1 or above or 2016 or above. Plus 2 GB of available disk space for the initial installation. |
2 Prerequisites

Required Components

- Security
  - Request Filtering
  - Basic Authentication
  - IP and Domain Restrictions
  - Windows Authentication
- Application Development
  - .NET Extensibility 4.5
  - ASP.NET 4.5
  - ISAPI Extensions
  - ISAPI Filters
- Management Tools
  - IIS Management Console
  - IIS Management Scripts and Tools
  - IIS 6 Metabase Compatibility
- Background Intelligent Transfer Service (BITS)
- IIS Server Extension
- Remote Server Administration Tools
  - BITS Server Extension Tools
- Windows Powershell 3.0 or above
- Microsoft Core XML Services (MSXML) 6.0
- Microsoft .NET Framework 4.5 Full
- Microsoft Visual C++ 2010 Redistributable package - only applicable for Application Manager
- Microsoft Visual C++ 2013 Redistributable package

Note
Application Manager requires both the x86 and x64 Redistributable packages to be installed on 64-bit operating systems.

- SQL Express 2014 SP1 - only applicable for server products
- IIS URL Rewrite Module 2 - only applicable for Personalization Server

Note
IIS URL Rewrite Module 2 has a dependency on IIS. IIS is automatically installed though the DesktopNow Installer for server products,

- XML Lite 1.0.1018.0
Install

In this Section:
- Installing AppSense DesktopNow on page 28
- AppSense DesktopNow Installer on page 29
- Evaluation Installation on page 29
- Advanced Installation on page 37
- Clients and Consoles Only Installation on page 47
- Manual Installation on page 54
- Post Installation Checklist on page 56
Installing AppSense DesktopNow

DesktopNow components can be installed using either the AppSense DesktopNow Installer or manually by using the individual MSIs.

Products can be installed with the Management Center to create integrated enterprise-scale solutions or installed as a standalone product aimed at evaluations.

The AppSense DesktopNow Installer provides a comprehensive process for installing any combination of AppSense products in a single fully integrated sequence. The installer performs a complete check for system prerequisites and provides you with the option of installing required technologies automatically.

See section AppSense DesktopNow Installer on page 29.

Alternatively, you can install each of the product components manually, by running the product installer packages for each component.

See section Manual Installation on page 54.

Caution

When installing AppSense products manually, you must ensure that all required technologies and AppSense components are added. A list of required technologies and AppSense components is available in the Prerequisites on page 22.

If using Active Directory, ensure that you follow Microsoft’s Active Directory Best Practices to enable it to work well with the Management Center Membership Rules and Deployment Groups.

Packages

Installer packages for each component in the AppSense DesktopNow product set include 32-bit and 64-bit versions of the Agent and Console, and for Environment Manager the PersonalizationServer.

Additional prerequisite third-party software components are provided with the installation media and can be installed automatically via the DesktopNow Installer or manually by running the relevant packages provided.
AppSense DesktopNow Installer

There are three types of DesktopNow installation:

- Evaluation - Install and configure all databases, services, and consoles automatically on one Windows server. See Evaluation Installation on page 29.
- Advanced - Provides control over which databases, services, and consoles install on each server. See Advanced Installation on page 37.
- Clients and Consoles Only - Install selected client agents and consoles. See Clients and Consoles Only Installation on page 47.

Evaluation Installation

1. Run the DesktopNow Installer by executing setup.exe from the installation media. The Welcome screen displays.
2. Click **Next** to display the License Agreement screen.
3. Read the AppSense End User License Agreement. If you agree to the terms, select **I accept the terms in the License Agreement** and click **Next**.

The Installation Type screen displays.

4. Select **Evaluation** and click **Next**.

**Note**

If SQL Server Express is already installed, you must have sysadmin permissions and SQL authentication enabled to select Evaluation installation type.
The SQL Server Installation screen displays.

5. Read the Microsoft Software License Terms. If you agree to the installation of SQL Server Express and the license terms, select **I accept the terms in the License Agreement**. Click **Next**.
If there are any prerequisites missing the Prerequisites screen displays.
6. The Prerequisites screen lists the prerequisites that are not currently installed. Select **Install All** to automatically install all missing prerequisites.

7. Once all of the components are successfully installed, the Installation Directory screen displays.

8. The default installation directory is C:\Program Files\AppSense. To continue the installation to the default location, click **Next**. Alternatively, **Browse** to select a new installation location.

9. Click **Install** to start the installation process.
The Product Install screen displays the progress of the installation.

10. If there are missing server prerequisites the Server Prerequisites screen displays.
11. Click **Install All** to install all the components listed.

12. Once all AppSense Product components have been installed, the Install Complete screen displays.

13. Click **Finish** to exit the Installer and open the User Assistance Portal.

   If you want to exit without opening the User Assistance Portal, deselect **Open User Assistance Portal on finish** and click **Finish**.
Advanced Installation

1. Run the DesktopNow Installer by executing setup.exe from the installation media. The Welcome screen displays.
2. Click **Next**.

The License Agreement screen displays.

3. Read the AppSense End User License Agreement. If you agree to the terms, select **I accept the terms in the License Agreement** and click **Next**.
The Installation Type screen displays.

4. Select **Advanced** and click **Next**.

The Product Selection screen displays.
5. Expand the Products to see all the product components that can be selected for installation.

**Note**
The Server components do not display if the Operating System is not compatible. Only the Product Consoles can be selected for install.

6. Select all the product components that you want to install.

If you want to install multiple instances of the Personalization Server or Management Server:

- Select **Add new Personalization/Management Server instance**. The Add New Instance dialog displays.
- Enter the name of the additional instance.
- Click **OK**. All instance names must be unique.

The server instance now displays in the list of components for selection. You can add up to 16 instances.

**Tip**
If you select Back on the Installer a message displays Remove Pending Instance [instance name]?

Click **OK** to continue going back and to remove the instance or click **Cancel** to remain on the Product Selection screen and keep the instance.

For multi-instance environments, further configuration is required, refer to the section **Extra Configuration for Multi-instance Personalization Servers** on page 46.
7. Click **Next**.

The Summary screen displays.
8. A summary of the products selected to install displays. Click **Next**.

   The SQL Server Installation screen displays.

9. Select whether you want to install a local SQL instance. The default selection is No.
   If you select **Yes**, the Microsoft Software License Terms display. Read the terms and if you accept them, select **I accept the terms in the License Agreement**.
   Alternatively, to use a remote SQL instance, accept default selection **No**, and click **Next**.
If there are any prerequisites missing, the Prerequisites screen displays.
10. The Prerequisites screen lists the prerequisites that are not currently installed. Select **Install All** to automatically install all missing prerequisites.

11. Once all of the components are successfully installed, the Installation Directory screen displays.

12. The default installation directory is C:\Program Files\AppSense. To continue the installation to the default location click **Install**. Alternatively, **Browse** to select a new installation location. Once you have selected the installation, click **Install**.
The Product Install screen displays the progress of the installation.

13. If there are missing server prerequisites the Server Prerequisites screen displays. Click Install All to install all the components listed.
14. Once all AppSense Product components have been installed, the Install Complete screen displays.

15. Click on the server hyperlink; http://<SERVER>:7750, to display the Server Configuration Portal in a web browser.

   Use the portal to configure the Management and Personalization servers, server instances and databases.

   **Note**
   See the [Server Configuration Portal Help](#) for further details.

16. Click **Finish** to exit the Installer and open the User Assistance Portal.

   If you want to exit without opening the User Assistance Portal, deselect **Open User Assistance Portal on finish** and click **Finish**.

**Extra Configuration for Multi-instance Personalization Servers**

When a named instance of the Personalization Server is installed, server self-registration is disabled for non-default instances. The new server must be manually added to the appropriate site using the Sites node in the Environment Manager Console.

**Note**
For further information, see the [Sites](#) topic of the [AppSense Environment Manager Help](#).
Clients and Consoles Only Installation

1. Run the DesktopNow Installer by executing setup.exe from the installation media. The Welcome screen displays.

2. Click **Next** to display the License Agreement screen.
3. Read the AppSense End User License Agreement. If you agree to the terms, select **I accept the terms in the License Agreement** and click **Next**.

The Installation Type screen displays.
4. Select **Clients and Consoles Only** and click **Next**. The Product Selection screen displays.

5. Expand the Products to see all of the product components that can be selected for installation.

**Note**

If you select a component that requires a reboot a warning message displays with the option to continue or cancel.
6. Once you have selected all the required products, click **Next**. The Summary screen displays.
7. Click **Next**.
If there are any prerequisites missing, the Prerequisites screen displays.
8. The Prerequisites screen lists the prerequisites that are not currently installed. Select **Install All** to automatically install all missing prerequisites.

9. Once all of the components are successfully installed, the Installation Directory screen displays.

The default installation directory is C:\Program Files\AppSense. To continue the installation to the default location, click **Install**. Alternatively, **Browse** to select a new installation location. Once you have selected the location, click **Install**.
10. Once all AppSense Product components have been installed, the Install Complete screen displays.

11. Click **Finish** to exit the Installer and open the User Assistance Portal.
   
   If you want to exit without opening the User Assistance Portal, deselect **Open User Assistance Portal on finish** and click **Finish**.
   
   If you selected products that require a reboot, you need to manually reboot.
# Manual Installation

The table below shows the list of the Windows Installer Packages (MSI) for each of the components in the AppSense DesktopNow suite, which you can run manually on the host computers. The list is organized per product and includes details about which components require a reboot of the host computer after installation.

## Note
When installing AppSense products manually, you must ensure that all required technologies and AppSense components are added. A list of required technologies and AppSense components is available in the Prerequisites on page 22.

<table>
<thead>
<tr>
<th>Installation File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DesktopNow</strong></td>
<td></td>
</tr>
<tr>
<td>ApplicationManagerConsole32.msi</td>
<td>Installs the DesktopNow console for creating configurations to deploy to managed computers hosting the agent.</td>
</tr>
<tr>
<td>ApplicationManagerConsole64.msi</td>
<td></td>
</tr>
<tr>
<td>ApplicationManagerAgent32.msi</td>
<td>Installs the Application Manager agent on managed computers. When a configuration is installed, the agent implements the configuration rules.</td>
</tr>
<tr>
<td>ApplicationManagerAgent64.msi</td>
<td></td>
</tr>
<tr>
<td>ApplicationManagerWebServices32.msi</td>
<td>Installs the AM Web Service used for Privilege Discovery to monitor applications that use administrative privileges to run. When installed and configured, the service collates data and allows you to create configurations based on the Privilege Discovery Results report.</td>
</tr>
<tr>
<td>ApplicationManagerWebServices64.msi</td>
<td></td>
</tr>
<tr>
<td>PerformanceManagerConsole32.msi</td>
<td>Installs the Performance Manager Console for creating configurations to deploy to managed computers hosting the agent.</td>
</tr>
<tr>
<td>PerformanceManagerConsole64.msi</td>
<td></td>
</tr>
<tr>
<td>PerformanceManagerAgent32.msi</td>
<td>Installs the Performance Manager agent on managed computers. When a configuration is installed, the agent implements the configuration rules.</td>
</tr>
<tr>
<td>PerformanceManagerAgent64.msi</td>
<td></td>
</tr>
</tbody>
</table>
| EnvironmentManagerConsole32.msi | Installs the Environment Manager Policy and Personalization individual or combined consoles for:  
| EnvironmentManagerConsole64.msi |   * Creating configurations to deploy to managed computers hosting the agent  
| Note: See Environment Manager Console Variants for further console information |   * Configuring the Personalization database.  
| EnvironmentManagerAgent32.msi | Note: For users with a Support Console role, the console opens in read-only mode. For further information, see the Environment Manager Support Console topic in the Help. |
| EnvironmentManagerAgent64.msi |             |
| EnvironmentManagerTools32.msi | Installs the Environment Manager Administrative Tools, which are a range of standalone tools to assist administrators when working with the Personalization Database and creating configurations. The tools run independently from Environment Manager and all other AppSense products. |
| EnvironmentManagerTools64.msi |             |
| PersonalizationServer64.msi | Installs the Personalization Server, which synchronizes user personalization settings between the SQL database and the managed computer. Must be configured using the Server Configuration Portal. |
| EnvironmentManagerTools32.msi |             |
| EnvironmentManagerTools64.msi |             |
Environment Manager Console Variants

There are three variants of the Environment Manager console:

- **Personalization** - Installs only the personalization element of Environment Manager
- **Policy** - Installs only the policy element of Environment Manager
- **Both consoles** - Installs the combined console; both personalization and policy are installed.

When you use the DesktopNow Installer to install Environment Manager, the combined Policy and Personalization Console is installed. Administrators may not require access to both. For example, they may only be responsible for configuring personalization and have no need for the policy side of the console. If installing the console manually via the EnvironmentManagerConsole.msi you have the option to install the Personalization or Policy console.

**Manually Install an Environment Manager Console**

1. Double-click the installer appropriate to the operating system:
   - EnvironmentManagerConsole32.msi
   - EnvironmentManagerConsole64.msi
2. On the Welcome screen, click **Next**.
3. Read the license agreement, if you accept the terms, select **I accept...** and click **Next**.
4. On the Destination Folder screen, click **Next**.
5. On the Console Features screen, select the console you want to install:
   - **Personalization**
   - **Policy**
   - Select both options to install the combined console (this is the default setting)
6. Click **Next**.
7. On the Ready to Install screen, click **Install**.
8. When the Install Complete screen displays, click **Finish** to exit the installer.

**Manually Adding Product Consoles to the Management Server Downloads page**

To manually add MSI or MSP files to the Management Server Downloads page, follow the following procedure:

1. Browse to:
   
   C:\Program Files\AppSense\Management Center\Server\Web Site\Downloads

2. Select the appropriate product folder, for example Application Manager.
3. Select the appropriate version folder, for example 8.7.0.0, or create it if it doesn’t exist.

Copy the MSI or MSP files into the folder.

**Post Installation Checklist**

Once you have installed AppSense DesktopNow using the Installer, check you have the following:

<table>
<thead>
<tr>
<th>Advanced</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Console</strong></td>
<td>Go to the <strong>Start</strong> menu and check <strong>AppSense &lt;product&gt; Console</strong> is present.</td>
</tr>
<tr>
<td><strong>Agent</strong></td>
<td>Run the <strong>AppSense Server Configuration Portal</strong> to create the database and upload the 32-bit and 64-bit agents. To check the agents are present, go to the <strong>Management Center</strong> console and check they are listed under <strong>Packages</strong>.</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>Open the console and select the <strong>Application</strong> menu button. You must save the blank configuration to implement the default rules. <strong>Select Save As &gt; Configuration in the Management Center</strong>.</td>
</tr>
<tr>
<td><strong>AM Web Services</strong></td>
<td>Open a browser and enter the following: <a href="http://localhost/AmAnalysisservice/status">http://localhost/AmAnalysisservice/status</a> <strong>Note:</strong> &lt;Localhost&gt; is replaced with the name of the machine that the service resides on. If the service is installed correctly a service website will be displayed.</td>
</tr>
</tbody>
</table>
Deployment Agent (CCA)

In this Section:
- Deployment Agent (CCA) on Managed Computers on page 58
- Installing the Deployment Agent (CCA) on page 58
- Integrated Install Deployment Agent Functionality on page 58
- Install Deployment Agent (CCA) Manually on page 60
- Install Deployment Agent (CCA) in Silent Mode on page 62
Deployment Agent (CCA) on Managed Computers

The Deployment Agent (CCA) is installed on managed computers to manage communications between the product agent and the AppSense DesktopNow.

The Deployment Agent (CCA) communicates with the Management Server to manage the download and installation of agents, configurations, and software package updates, and also sends event data generated by the product agents to the Management Server.

Installing the Deployment Agent (CCA)

Install the Deployment Agent (CCA) on all computers to be managed by the AppSense DesktopNow. The Deployment Agent (CCA) can be distributed using the integrated Install Deployment Agent functionality in the Management Console, by downloading the ClientCommunicationsAgent32/64.msi package from the Management Server website or by third-party deployment mechanisms.

It is recommended you set up Membership Rules and Deployment Groups in the Management Console before installing the Deployment Agent (CCA). Refer to the AppSense Management Center Help for further details.

The Install Deployment Agent functionality in the Management Console can be run in small and medium scale enterprise environments to deploy the Deployment Agent (CCA) to multiple computers, or to repair or modify the URL path for currently deployed Deployment Agents to change the http or https prefix and port number.

IT administrators in organizations often create master images that include the operating system with all the required software and updates required for a new computer, as a labor-saving approach to setting up multiple computers. It is recommended to install the Deployment Agent (CCA) on a master image prior to rolling out to computers in your organization.

Integrated Install Deployment Agent Functionality

The Management Console provides an Install Deployment Agent function that allows you to deploy the Deployment Agent (CCA) to multiple computers that match the DesktopNow Deployment Group and Membership Rules. The agent can be deployed either on a Microsoft Active Directory network or in a Microsoft Windows workgroup in small or medium scale environments.

Workflow

The Install Deployment Agent functionality detects the DesktopNow deployment groups and uses group membership rules to provide the list of computers to which the Deployment Agent (CCA) can be deployed. Active Directory is queried for active directory types for membership rules by computer, groups, and containers. Alternatively, you can manually include or exclude computers from the list by NetBIOS Name.

Note

The Install Deployment Agent functionality can only deploy the Deployment Agent (CCA) to computers that are members of Deployment Groups configured in the DesktopNow console.
The software requirements for the target client computers are detected and the latest 32-bit or 64-bit version of the Deployment Agent (CCA) installation package is downloaded. Packages are distributed to the target computers and installed silently with the correct URL of the Management Server.

The basic steps required to install the Deployment Agent (CCA) are as follows:

**STEP 1  SET UP CLIENT ACCESS CREDENTIALS**
1. Navigate to Home > Global Settings > Access Credentials tab.
2. Enter the user credentials (user name and password) for an account on the computer on which the Deployment Agent (CCA) is being installed. The account must have local administrator privileges.

You can add multiple accounts. They will be attempted in the order in which you list them.

<table>
<thead>
<tr>
<th>Caution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You will not be able to install the Deployment Agent (CCA) on any endpoint using the integrated Install CCA functionality if the Access Credentials have not been set up.</td>
</tr>
</tbody>
</table>

**STEP 2  CONFIGURE SETTINGS FOR THE DEPLOYMENT GROUP**
2. Create a New Deployment Group.
3. Select Settings for the new deployment group. The Settings work area opens on the General tab.
4. In the Server Polling and Downloads section, specify how often the client computers check for and download new agents or configurations. Use the slider to set the poll variance. The poll variance reduces the impact of multiple clients polling and downloading at the same time.
5. In the Event Data Uploads section, specify how often event data is uploaded to the Management Server. Use the slider to set the upload variance. The upload variance reduces the impact of multiple clients uploading data at the same time.
6. In the Deployment Agent Permissions, specify whether the deployment agent can self-register, unregister, or make agent and configuration updates outside of the set installation schedule.
7. Select the Installation tab.
8. Set up the agent and configuration installation schedules for the deployment group.
STEP 3  CREATE MEMBERSHIP RULES FOR THE DEPLOYMENT GROUP

   
   Each Deployment Group has a one to one relationship with a set of Membership Rules. The Membership Rules act like a filter to discover computers in Active Directory.

2. In the Actions panel, select **Edit Conditions** to add a new condition based on NetBIOS Name or Active Directory.

3. Select **Submit** from the Membership Rules work area.

4. In the Actions panel, select **Discover**.
   
   The discovered computers that match the Membership Rules are listed in the relevant Deployment Group > Computers node.

   **Note**
   
   For the computers discovered by Membership Rules, the Computer Status should initially display: *No CCA deployed.*

STEP 4  INSTALL CCA

1. Navigate to **Home > Deployment Groups > [Deployment Group] > Computers**

2. Select the computer or computers on which you want to install the Deployment Agent (CCA).

3. In the Actions panel, select **Install CCA**.
   
   The Client Access Log provides details on the installation progress. The Deployed (%) column indicates the percentage of the package deployed.

Install Deployment Agent (CCA) Manually

You can manually install the Deployment Agent (CCA) on a managed computer by downloading and running the ClientCommunicationsAgent32/64 installation package on a client computer.

1. Launch a web browser and navigate to the Management Server website at the following address:

   https://[computer name]/ManagementServer

   **Note**
   
   If you have not configured SSL communications, use the HTTP prefix for the Management Server website:

   http://<computer name>/ManagementServer/
The Management Center download page displays.

Note
- The Downloads page is best viewed in Internet Explorer 8 or higher
- Make a note of the Management Server URL displayed in the download page.
- You can also download: Management Console, EULA, Release Notes, Application Manager, Environment Manager and Performance Manager Consoles, and Prerequisites Software.

2. Download and run the appropriate 32-bit or 64-bit ClientCommunicationsAgent installation MSI package.
3. In the Deployment Agent (CCA) installation Welcome screen, click Next.
4. In the License Agreement screen, read the license agreement. If you accept the terms, select and click Next.
5. In the Installation Directory screen, leave the default installation directory unchanged, and click Next.
6. The Settings screen displays.
   Enter the Management Server computer name, http://<Computer Name>/.

Note
If you have configured SSL communications, use the HTTPS prefix for the Management Server website:
https://<computer name>/
7. Click **Next** to proceed.
8. When the installation is complete, click **Finish** to exit the installation wizard.

You have now successfully installed the AppSense Deployment Agent (CCA). The host computer is able to connect to your Management Server, ready to download product agents, license, and configuration software packages according to the settings configured for the deployment group to which the current computer belongs.

### Install Deployment Agent (CCA) in Silent Mode

You can install the AppSense Deployment Agent (CCA) silently via a third-party deployment mechanism or from a command prompt.

**Note**
Use the HTTPS prefix for the Management Server website only if you intend to install an SSL certificate on the server computer and managed computers are located in the same Active Directory domain as the Management Server.

`msiexec.exe /qn /i "<MSI file path>\CommunicationsAgent.msi"`  
`WEB_SITE="https://<Management Server Name>/" GROUP_NAME="<DeploymentGroup>"`

- `/i` — Install
- `/qn` — Quiet mode install without the user interface.
- `WEB_SITE` — Enter the Management Server website address using the name of the host computer.
- `GROUP_NAME` (optional) — Enter the Deployment Group name to which the Deployment Agent (CCA) should register. The Deployment Agent can only register with a group that is set up to allow the agent to self-register. Otherwise, the agent attempts to register with the Management Server deployment groups according to group membership precedence:
  - `GROUP_NAME` self-register
  - Deployment Groups - membership rules
  - (Default)Group – if no match is found

**Allow self-registration with this group**

You can set up a deployment group to allow the Deployment Agent (CCA) to self-register with a specific group when installed using the command line.

This option is disabled by default but provides an alternative method for installing the deployment agent on managed endpoints to register with a specific Deployment Group on the Management Center rather than predefining the group membership in the Management Console.

Enable **Allow self-registration** in the Deployment Agent Permissions section on the General tab in the Settings work area for the relevant Deployment Group.
Licensing

In this Section:
- About AppSense DesktopNow Licensing Console on page 64
- Managing Licenses on page 64
About AppSense DesktopNow Licensing Console

License details are included in the License Agreement which is issued when an order for AppSense software has been completed.

The License Agreement includes the following information:

- Product, Feature, and Version Details
- Issue Date
- Expiry Date
- Customer Name
- Serial ID

Together with the license agreement you will receive either a TXT file or a LIC file. Use these in the AppSense Licensing Console to add or import the license.

Note
For information about managing licenses in the Management Center see the AppSense Management Center Help.

Use the AppSense DesktopNow Licensing Console to:

- Manage licenses for DesktopNow products.
- Export license packages to MSI or LIC file format for saving to the AppSense Management Center or other computers which can be remotely accessed.
- Import and manage licenses from LIC file format.

When the Licensing Console is launched, all the current licenses display.

Managing Licenses

The following procedures describe how to add a new license, import and export licenses or to backup a set of licenses.

 pomięd 1

Add a License

1. Open the AppSense Licensing console.
2. Click Add.
   The Add License Key dialog displays.
3. Enter the license key and click Add.
   If you received a TXT file from AppSense, open the file and copy the license key, paste it in to the Add License Key dialog.
   If you received a LIC file from AppSense, refer to the Import a License File section.
   Details of the license are displayed in the console and the license key is added to the following location:

%ALLUSERSPROFILE%\AppSense\Licenses
**Activate a License**
1. Once a license is added, you may need to activate it, depending on the license type. Click **Activate**.
2. Type or copy and paste the activation code.
3. Click **Enter** to accept the code.
4. Close the **Licensing** console. The settings are automatically saved.

**Import a License File**
1. Open the AppSense Licensing console.
2. Click **Import** to display the Windows Open dialog.
3. Select the required license LIC file.
4. Click **Open**.
   Details of the license are displayed in the console and the license key is added to the following location:
   `%ALLUSERSPROFILE%\AppSense\Licenses`

**Export a License File**
1. Open the AppSense Licensing console.
2. Highlight the license you want to export.
3. Click **Export** to display the Windows Save As dialog.
4. Browse to the required location to save the license file.
5. Enter a name for the file.
6. Select the file type: MSI or LIC.
7. Click **Save**.
   A file is created and saved in the selected location. This file can be copied to any network location and loaded via the AppSense DesktopNow Suite Licensing console or in the Management Center console.

**Remove a License**
1. Highlight the required License and click **Remove**.
2. A confirmation dialog displays, click **Yes** to confirm.
   The selected license is deleted and removed from the console and the MS Windows registry.
Server Configuration

In this Section:
- About Server Configuration Portal on page 67
- DesktopNow Summary on page 70
- Management or Personalization Summary on page 71
- Export Scripts on page 85
- Server Upgrades on page 86
- Database Updates on page 89
- Technical Reference on page 90
About Server Configuration Portal

The Server Configuration Portal provides the functionality to:

- Connect Personalization Servers or Management Servers to existing databases.
- Create new databases.
- Configure the DesktopNow Servers.

Multiple Personalization Servers and Management Servers can be connected to the same database and managed via the Server Configuration Portal.

The Server Configuration Portal is available from a web browser on any Windows Server where a DesktopNow Server has been installed. The default location is http://<servername>:7750/

Tip

The Server Configuration Portal drives a rich set of PowerShell cmdlets. For further details refer to the AppSense Server Configuration Portal Scripting Guide.

Separate Websites for each DesktopNow Server Component

Unless upgrading existing 8.x installations, the Server Configuration Portal, Personalization Server, and Management Server install into their separate IIS websites, named Configuration, Personalization, and Management respectively. Each website can be configured with two IIS bindings: one using port 80 with a host header, the other using a different port without a host header, see default port list below.

If you do not input a host header entry during the creation of an instance, just one binding is created. For example, for a single Management Server instance, a binding of 7751 is created.

Host headers allow multiple websites to share the same port and are reachable by configuring DNS records on a DNS server. If a host header is chosen that includes the fully qualified domain name of the domain the IIS Server is joined to, it's only necessary to create a CNAME record for the DNS server.

Access to any of the DesktopNow server components, before configuring DNS records, must be through the following ports:

- Configuration: 7750
- Management: 7751
- Personalization: 7771
**Alternative IIS Configuration**

If the default website has been removed, disabled or is not being used, you can use any of the DesktopNow server components on port 80 without a host header. By removing the default website, or stopping it and giving it a dummy host header, it's possible to remove the host header from one of the DesktopNow server components, effectively making this the new default website. You can do this only when the Management server and Personalization server are installed on different IIS servers.

Any changes made to the IIS settings via the IIS Manager are reflected in the Server Configuration Portal.

You may want to change the website to configure the port, the bindings to IP addresses, the delegation of features, or SSL certificate for a given Management Server.

**Database Accounts**

The Server Configuration Portal uses two accounts: the Configuration account and the Service account. Both are set up by the database administrator.

Accounts can be added, edited, or removed. Once an account is added, it is assigned to all services.

**Configuration Account**

The Configuration Account which is used to connect to the database to perform operations, including creating, upgrading and configuring the Management Server and database. The account is used to perform the following tasks:

- Create the database - only performed if the database does not exist, requires dbcreator rights.
- Create logins - only performed if a login does not exist, requires security_admin rights.
- Ensure the database schema matches the version defined by the product.
- Check for variances - whether the properties of the database match the product expectations.
- Confirm the database user logins.
- Populate the initial data set into the database.

The Configuration account must have dbo rights, or be a member of the ManagementServerAdministrator role. Some additional rights may be needed for optional tasks, which are detailed in the list above.

The account can use either Windows Authentication, that uses the account currently running the Server Configuration Portal, or SQL Authentication.
Service Account

The Service account is used by the Windows services and web services that make up the Management Server. This role has access to all of the Management Server stored procedures.

The Server Configuration Portal persists the username and password of the Service account within the FileName.exe.config and web.config files of the Windows and web services. Both the username and password are encrypted using the Microsoft Crypto API using the certificate of the local machine.

The Service account must be a member of the ManagementServerService role and should not have any additional rights on the database of the SQL instance. The account can use either Windows Authentication or SQL Authentication.

Note
For further details on Services refer to Technical Reference on page 90.

Administrator Privileges

The Server Configuration Portal must be run with elevated privileges. The user running the portal must have administrator rights on the server being administered. It may be the case that the user has administrator rights to the server, but not to the SQL server. In this case, you can use PowerShell to export the SQL Scripts that need to be run to create and configure the database.

Note
For further details on PowerShell cmdlets refer to the AppSense Server Configuration Portal Scripting Guide.
Launch the Server Configuration Portal in a web browser. The default location is http://<servername>:7750/

The DesktopNow node displays the DesktopNow Summary page showing an overview of the Management and Personalization servers and databases represented by colored tiles.

Database Tile
Displays the total number of databases per product.

- Blue - no variances.
- Red - a database needs one of the following:
  - Upgrading - the database schema is out of date.
  - Updating - the database data is out of date.
  - Configuring - configurer details missing.

Click the red tile to go to the first database that needs attention in the tree structure. The node in the tree structure also displays in red.

Once that database variance has been fixed, go back to the DesktopNow Summary page, if the tile remains red there are further databases that need attention. Fix all variances until the tile turns green, which indicates all databases are up to date and configured correctly.

Servers Tile
Displays the total number of servers per product. The server tile is always blue.
Server Instances Tile
Displays the total number of instances for all servers per product. The server instance tile is always blue.

Unconfigured Server Instance Tile
Displays the number of unconfigured server instances for all servers per product.
- Green - no unconfigured instances.
- Red - one or more instances need configuring.

Click the red tile to go to the first instance that needs attention in the tree structure. The node in the tree structure also displays in red.

Once the instance has been configured, go back to the DesktopNow Summary page, if the tile remains red there are further instances that need configuring. Fix all variances until the tile turns green which indicates all instances are configured correctly.

Management or Personalization Summary
The Management or Personalization nodes display the product Summary page, where you can see an overview of the servers and databases for the product.

Click any tile to go to that node in the tree structure.
Database Tile
Displays the total number of databases for the product.
- Blue - no variances.
- Red - a database needs one of the following:
  - Upgrading - the database schema is out of date.
  - Updating - the database data is out of date.
  - Configuring - configurer details missing.

Click the red tile to go to the first database that needs attention in the tree structure. The node in the tree structure also displays in red.

Once that database variance has been fixed, go back to the product Summary page. If the tile remains red there are further databases that need attention. Fix all variances until the tile turns green, which indicates all database are up to date and configured correctly.

Servers Tile
Displays the total number of servers for the product. The server tile is always blue.

Server Instances Tile
Displays the total number of instances for all servers for the product. The server instance tile is always blue.

Unconfigured Server Instance Tile
Displays the number of unconfigured server instances for all servers for the product.
- Green - no unconfigured instances.
- Red - one or more instances need configuring.

Click the red tile to go to the first instance that needs attention in the tree structure. The node in the tree structure also displays in red.

Once the instance has been configured, go back to the product Summary page, if the tile remains red there are further instances that need configuring. Fix all variances until the tile turns green which indicates all instances are configured correctly.
Database

The Database node displays the product Database Summary page, where you can see a list of all databases and create or delete them.

Servers and Databases

Create a New Database

You can create databases for both the Management and Personalization servers.

1. Select the required node:
   - **Management > Management Databases**
   - **Personalization > Personalization Databases**

   The product Database Summary page displays in the work area.
2. Click **CREATE NEW**.

   The New Database Connection page displays.

3. Enter a **Friendly Name** for the database.

4. Click in the **Server Name** field to display a drop down of all known servers. Select the required server.

5. Enter the **Username** and **Password** for the Configuration account.
   
   To create a new database, ensure the configuration account has dbcreator server privileges and enter a unique database name.
   
   To set up the schema on a new empty database, ensure the configuration account is the database owner or a member of the db_owner role, and select the database from the list.
   
   To upgrade an existing database, the configuration account must have dbo privilege, and the database must be selected from the list. Always backup your database before performing an upgrade.
   
   To use an existing database, the configuration account must be a member of the ManagementServerAdministrator or dbo database roles.

   **Tip**
   
   For further details on the Configuration and Service accounts refer to the section **Database Accounts** on page 68.
6. Select the Authentication Type:
   - **Windows Authentication**
     Pass-through authentication where the credentials of the currently logged on domain user are automatically provided to access the database.
   - **SQL Authentication**
     Specify an SQL Authentication account to provide access to the database. Accounts, including both username and password, are created in the SQL Server itself rather than making use of existing Windows domain accounts.

7. Click **CHECK** to validate the credentials.

8. Click in **SQL Database Name** to display a dropdown of all known databases. Select the required database. Or enter a new name to create a new one.

9. Enter the **Username** and **Password** for the Service Account.
   The Web service and Windows services use these credentials for the database connection.

10. Select the Authentication Type:
    - **Windows Authentication**
      A Windows username and password must be supplied each time access to the database is required.
    - **SQL Authentication**
      Specify an SQL Authentication account to provide access to the database. Accounts, including both username and password, are created in the SQL Server itself rather than making use of existing Windows domain accounts.

     **Note**
     If the Service account does not already exist in the SQL Server and the Configuration account has **securityadmin** server privileges, a new account is created.

11. Click **CHECK** to validate the credentials.

12. Click **CREATE** to start the database creation.
    Once created, the database displays in the tree structure under the relevant Database node.
**Edit a Database Connection**

You can edit databases for both the Management and Personalization servers.

1. Select the required node:
   - **Management > Management Databases**
   - **Personalization > Personalization Databases**

   The product Database Summary page displays in the work area.

2. Select the database node that you want to change. The database detail page displays in the work area.

<table>
<thead>
<tr>
<th>Servers and Databases</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Database Connections" /></td>
<td></td>
</tr>
<tr>
<td><strong>Management Database</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Database Details" /></td>
<td></td>
</tr>
<tr>
<td><strong>Personalization Database</strong></td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="Other Databases" /></td>
<td></td>
</tr>
</tbody>
</table>

- **Management Database**
  - Server Name: `local\SQLExpress` Management\Server\_d\6x6IP11`  
  - SQL Database Name: `Management\Server\_d\6x6IP11`  
  - Configuration Account: `deslopros\admin`  
  - Service Account: `deslopros\admin`  
  - Authentication: `Windows`  
  - Password: `**********`  
  - Authentication: `SQL`  

- **Personalization Database**
  - Server Name: `local\SQLExpress` Personalization\Server\_d\9y8P8B`  
  - SQL Database Name: `Personalization\Server\_d\9y8P8B`  
  - Configuration Account: `deslopros\admin`  
  - Service Account: `deslopros\admin`  
  - Authentication: `Windows`  
  - Password: `**********`  
  - Authentication: `SQL`
3. Amend the user name for the Configuration Account.
4. Enter the **Password**.
5. Select the Authentication Type:
   - **Windows Authentication**
     Pass-through authentication where the credentials of the currently logged on domain user are automatically provided to access the database.
   - **SQL Authentication**
     Specify an SQL Authentication account to provide access to the database.
     Accounts, including both username and password, are created in the SQL Server itself rather than making use of existing Windows domain accounts.

To upgrade an existing database, the configuration account must have dbo privilege, and the database must be selected from the list. Always backup your database before performing an upgrade.

To use an existing database, the configuration account must be a member of the ManagementServerAdministrator or dbo database roles.

6. Click **CHECK** to validate the credentials.
7. Amend the username and for the Service Account.
8. Enter the **Password**.
9. Select the Authentication Type:
   - **Windows Authentication**
     A Windows username and password must be supplied each time access to the database is required.
   - **SQL Authentication**
     Specify an SQL Authentication account to provide access to the database.
     Accounts, including both username and password are created within the SQL Server itself rather than making use of existing Windows domain accounts.

     If the Service account does not already exist in the SQL Server and the Configuration account has **securityadmin** server privileges, select **Create SQL Account** and a new account is created.

10. Click **CHECK** to validate the credentials.
11. Click **SAVE CHANGES** to save the details.
Server

The product Servers node displays the product Server Summary page, where you can see a list of all servers for the product. Click any server name to go to the Server Summary page for the selected server.

Servers and Databases

- DesktopNow
  - Management
    - Management Databases (2)
      - Local\(\backslash\)SQL\(\backslash\)Express\(\backslash\)Management\(\backslash\)Server\(\_\)6\(\_\)D\(\_\)P\(\_\)
      - ManagementDatabase
    - Management Servers (1)
- Personalization
  - Personalization Databases (1)
    - Local\(\backslash\)SQL\(\backslash\)Express\(\backslash\)Personalization\(\backslash\)Server\(\_\)DB\(\_\)MY\(\_\)M1\(\_\)
  - Personalization Servers (1)

Click a server to go to that server node in the tree structure and display the Server Summary page.
Server Summary

The Server node in the tree structure displays the Server Summary page in the work area where you can see a list of all instances for the selected server.

Click any Instance name to display the instance configuration page.
The status of each instance is listed together with whether logging is set and if there are any variances.

If there are no variances 'None Detected' displays. If there are variances the word 'Detected', displays in the Variances column. Click **Detected** to display the list of variances.

<table>
<thead>
<tr>
<th>ISSUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSite, SiteRunning: The Web Site must be running to be operable the World Wide Web Publishing Service may not be running. This must be started manually to fill this variances.</td>
</tr>
</tbody>
</table>

You can select to **FIX ALL** or **FIX SELECTED** variances. Click **REFRESH VIEW** to update the list and **DONE** when finished.
Server Instance

When you select the instance node, the page that displays depends on whether the instance has been configured.

For instances that are not configured, the Configure Instance page displays.

Configure an Instance

1. Click a server instance to display the Configure Instance page.
2. The Website name is either Management, Personalization, or Configuration. You can edit the Port number, the default port numbers are:
   - Management - 7751
   - Personalization - 7771
   - Configuration - 7750
3. Edit the Host Header and Port number or leave as the default.

Tip

The host header should contain your domain name only if you intend to add a CNAME record to DNS.
4. Select the **Authentication** type:
   - Windows authentication - Deployment Agents must authenticate with the server using Windows Authentication. This increases the security of the server, ensuring only computers in the domain can access the server.
   - Anonymous authentication - Deployment Agents can access the server unchallenged.

5. Select the **Database Connection** from the dropdown list or alternatively type in a name. If the database does not exist, a *CREATE NEW* option displays. Click it to create a new database connection.

6. Click **CREATE INSTANCE** to start the configuration process.
   Once the instance is configured, any red tiles relating to this instance are updated and the settings can be specified.

**Instance Settings**

1. Click a configured server instance in the tree structure. The Instance Settings page displays.

2. Set the instance Status. The instance can be Online or Offline.

3. Set Logging to Enabled or Disabled. Logging is disabled by default.
   Log files are written to the websites Bin folders.
4. Click **RECHECK** to run a variance check. If the instance has variances that need fixing, the text 'Detected' displays. Click on **Detected** to display the variances. If there are no variances the text 'None Detected' displays.

5. Click the URL to go to the website.

6. Set the Authentication type:
   - **Windows authentication** - Deployment Agents must authenticate with the server using Windows Authentication. This increases the security of the server, ensuring only computers in the domain can access the server.
   - **Anonymous authentication** - Deployment Agents can access the server unchallenged.

   **Note**
   If the Deployment Agent (CCA) is installed on computers in a Workgroup you must select **Anonymous authentication**.

7. To set the Database Connection, select from the dropdown list.

8. Click **UPDATE** to save any changes.

---

**Encryption**

If multiple Management Servers are being utilized in a failover scenario, the Encryption node is used to share the encryption key between each Management Server. Any encryption that is required uses the Microsoft Windows Cryptographic Service Provider. It is also used to back up the key securely in the database.

If failover servers are being used, all the servers must use the same public-private key pair.
First, a transfer key needs to be made available by the master server. The transfer key contains both the public and private keys. Click **STORE** to save the key in the database in a password protected format.

Once the password has been stored, the transfer key is shown as present and can now be retrieved by other servers to create the correct public-private key pair.

- **Server Configuration**
- **Management or Personalization Summary**

<table>
<thead>
<tr>
<th>Servers and Databases</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>DesktopNow Management</td>
<td>Encryption Keys are required when you are using more than one Management Server. The key must first be stored on one server and then retrieved on all other servers.</td>
</tr>
<tr>
<td>Management Databases 1</td>
<td>Encryption Key Status: Valid</td>
</tr>
<tr>
<td>SQL EXPRESS, ManagementServerوفقی رختی</td>
<td>CREATE</td>
</tr>
<tr>
<td>ManagementDatabase</td>
<td></td>
</tr>
<tr>
<td>Management Servers 1</td>
<td>Transfer Key Status: Present</td>
</tr>
<tr>
<td>DesktopNow</td>
<td>RETRIEVE</td>
</tr>
<tr>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Personalization</td>
<td></td>
</tr>
<tr>
<td>Personalization Databases 1</td>
<td></td>
</tr>
</tbody>
</table>
Personalization Operations

To navigate to the Personalization Operations node in the tree structure select, **DesktopNow > Personalization > Personalization Servers > [Server] > [Instance] > Personalization Operations**. The Settings page displays.

You can update the settings so that the website is Online or Offline, enable or disable logging, and specify the authentication type.

Export Scripts

AppSense Servers uses an SQL Server database to store data. The installation procedure requires sysadmin access to the SQL Server instance in order to create and initialize the database. When the person installing the Server does not have sysadmin access, scripts can be exported to enable the database to be set up. It is assumed that the SQL Server instance is on a separate machine to the AppSense Server.

**Note**

Refer to the *AppSense Server Configuration Scripting Guide* for further PowerShell details.
Server Upgrades

In-place upgrades

An in-place upgrade allows for the minimum service interruption, requires no change to the client configuration for either the Management Server or the Personalization Server, and allows for staged upgrades of either the Deployment Agent (CCA) or the EM Agent.

In-place upgrades of either the Management Server or the Personalization Server install into the default website or whichever website in which the Management Server or the Personalization Server is installed.

Caution
A Personalization Server requires access to the root of the website in which it is installed. If installed on the default website, it overwrites any existing applications that are currently using the root of the default website.

If you already have an application installed into the root of the default website, you should attempt an in-place upgrade of the Management Server only. The Personalization Server should be installed as a new installation into the Personalization website; refer to the New Server/Clean Install Upgrades on page 86.

New Server/Clean Install Upgrades

Uninstalling the existing DesktopNow servers and reinstalling or provisioning a new server provides an alternative upgrade strategy, but does include caveats. You can use a number of methods to upgrade the Management server or Personalization server. However, you must plans carefully because you may need to update the connection URLs for clients and console. The most common options following a new installation and database upgrade are:

- Remove the host header and make either the Management Server or the Personalization Server the default website. This is described in the section Alternative IIS Configuration on page 68.
- Update DNS records, Load Balancer, or Virtual IP to direct HTTP traffic to the new website. For information on DNS and the use of host headers, see Separate Websites for each DesktopNow Server Component on page 67.

Management Server only:

- Having upgraded the Management Server and database, deploy the version 10.0 Deployment Agent (CCA). This ensures the Deployment Agent (CCA) URL is updated.

  Note
  This method is particularly advantageous if you also intend to deploy a version 10.0 license.

- For staged upgrades of the Management Center, add the new Management Server URL to the Failover Server List prior to bringing the version 10.0 Management Server online. The Deployment Agent (CCA) continues to communicate with the 8.x Management Server until it is taken offline and the version 10.0 Management Server comes online.
Personalization Server only:
- Deploy a new Environment Manager Policy file containing the updated Personalization Server URL.
- For staged upgrades of Environment Manager Personalization, it is recommended to make use of the Virtual Hosts feature of Personalization along with updated DNS records prior to any upgrade.

Testing Server Upgrades

On completion of the Personalization Server upgrade, test that the server is functioning by visiting the PersonalizationServer website at:

http://localhost/PersonalizationServer/status.aspx

or

https://localhost/PersonalizationServer/status.aspx

If you are running the test from a remote location to the Personalization Server, replace localhost with the server name. After approximately 30 seconds, a page displays to confirm successful connection. Check the details are correct and the tests are successful. When connection is complete, the Personalization Server can accept requests from managed endpoint devices.

Replication Servers

Upgrades cannot be performed while replication is enabled. Replication must be disabled on the master and subscribers prior to upgrading all servers to the same version.

Upgrading to a newer release includes a database schema change. Therefore all personalization servers are locked out of the database until the upgrade process is complete.

Load Balanced Servers

When servers are load balanced and operate as individual entities using a common IP address, it is recommended that all servers are taken offline before upgrading. When the upgrade is complete, servers can be brought back online and added back to the network load-balanced configuration.

When upgrading servers, the first server to be upgraded also upgrades the database. This ensures that the server and database versions match after the schema change. During an update of a server and the database, all other servers are locked out.
Server Load Balancing Health Monitors

If using a load balancer monitor to check the server status page (status.aspx) where the server is set to Windows Authentication, the monitor must provide Windows credentials in the HTTP headers or the server will respond with an “unauthorized” reply. As an alternative, use one of the following URLs - appropriate to your setup - to test check the server status.

To check the health of load balanced servers use the following methods, appropriate to your setup.

IIS7 - Windows Server 2008 and R2 or greater use the following URLs in an Internet browser, replacing <SERVER> with the name of the required server:

  http://<SERVER>/PersonalizationServer/dbmonitor.aspx
  Checks the connection with the database. Returns “OK” (http status 200) if the database can be contacted and “FAIL” (http status 503) if it cannot.

  http://<SERVER>/PersonalizationServer/pingmonitor.aspx
  Checks whether the server address is reachable returning “OK” (http status 200) if successful. No response indicates an error.
Database Updates

Using the Server Configuration Portal

If the database is out of date, the Database Updates Required page automatically displays when you open the Server Configuration Portal.

<table>
<thead>
<tr>
<th>SERVER NAME</th>
<th>DATABASE NAME</th>
<th>CURRENT VERSION</th>
<th>LATEST VERSION</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>localSQL\Express</td>
<td>ManagementServer.MAN</td>
<td>8.1.0</td>
<td>10.0.0</td>
<td>Schema Upgrade Required</td>
</tr>
<tr>
<td>localSQL\Express</td>
<td>Personalization Server \p1\now\APP</td>
<td>8.6.7</td>
<td>10.0.0</td>
<td>Schema Upgrade Required</td>
</tr>
</tbody>
</table>

1. Select the databases you want to upgrade and click UPDATE SELECTED. Alternatively, select UPDATE ALL to upgrade all databases.

2. Once the Status changes to Upgrade Successful, click CLOSE to display the DesktopNow Summary page.
Technical Reference

Services

There are four associated services with the Management Center:

- **AppSense Alerts Service** - responsible for creating alerts based on events for the Management Server, and dispatches associated actions.
- **AppSense Events Dispatcher Service** - responsible for monitoring for new event files being uploaded and adds the events to the Management Server database.
- **AppSense Scheduler Service** - responsible for managing all scheduled tasks associated with the Management Server. This includes discovery and offline machine detection.
- **AppSense Deployment Service** - responsible for managing the installation of the Deployment Agent (CCA) when chosen by the user from the Management Console.

To generate diagnostic logs for Management Server Services set Logging to Enabled on the Instance Detail page in the Server Configuration Portal. The log files are stored in `%Program Files%\AppSense\Management Center\Server\Bin` by default.

Websites

**ManagementServer**

The ManagementServer root web directory hosts the Downloads web page for downloading the Management Console, Client Communications Agent, AppSense products, and documentation.

A diagnostics log can also be generated from this page. It is stored at `%Program Files%\AppSense\Management Center\Server\Web Site` by default.

**ManagementServer/Deployment**

The ManagementServer/Deployment web directory provides the Management Server web services that the Deployment Agent (CCA) uses to access the Management Center database. These hosted web services are:

- **Polling** - Managed endpoints receive settings such as poll periods and installation schedule during a poll.
- **Prerequisite checking & installation** - Managed endpoints download agents, configurations, and prerequisites using BITS.
- **Event Collection** - Managed endpoints upload the majority of events using BITS.
- **Server Diagnostics** - Managed endpoints send high priority events.

A diagnostics log, `DeploymentDirectory.log`, can also be generated from this page which is stored at `%Program Files%\AppSense\Management Center\Server\Web Site\Deployment` by default.

Note

For further details on log files, refer to the [Client Access Log](#) on page 60.
ManagementServer/ DataAccess
The ManagementServer/DataAccess web directory provides the interface to the Data Access Services. All communication from the Management Console comes here.

ManagementServer/ PackageManagement
The ManagementServer/PackageManagement web directory provides an interface to the Package Management Services. All communication from the Application Manager, Environment Manager and Performance Manager consoles comes here.

Securing Communications using SSL
You can optionally configure the Management Server website to support Secure Socket Layers (SSL) to provide secure communications using Active Directory.

SSL provides confidentiality and integrity of communications to ensure sensitive data is accessible only by authorized users, including:

- Event data
- Agents and agent configuration data

Note
If you are setting up SSL certificates on web servers using other supported operating systems and other versions of Microsoft SQL Server, see the following for further information:

Setup SSL on IIS
This section provides information about setting up the website for SSL by creating a self-signed certificate.

1. In **Start > All Programs > Administrative Tools > Internet Information Services (IIS) Manager**, select the [ServerName] node and in the IIS section click **Server Certificates**.
2. Ensure a suitable certificate is listed. If not, create or import a certificate using the options in the Actions panel.
3. Select the **Website** for the product and click **Edit Bindings** in the shortcut menu.
4. Click **Add** and in the **Type** drop-down list select **HTTPS**.
5. In the **SSL Certificate** drop-down list, select the certificate.
6. Click **OK** and **Close**.

**Note**
Other types of certificate issued by a trusted Certification Authority are also supported.

Database

SQL Server AlwaysOn is the preferred SQL Server technology to support High Availability scenarios and the DesktopNow servers have been optimized to support this technology.

**Note**
SQL mirroring is supported for versions 8.x. For more information, see the AppSense Management Center Install and Upgrade Guide for the relevant version. Visit [support.appsense.com](http://support.appsense.com) to download the relevant guide.

Guidance on configuring SQL Server AlwaysOn Availability Groups can be found here:
- Overview of AlwaysOn Availability Groups
- Prerequisites, Restrictions, and Recommendations for AlwaysOn Availability Groups
- Configuration of a Server Instance for AlwaysOn Availability Groups
- Creation and Configuration of Availability Groups

AppSense whitepaper on SQL Server High Availability and Disaster Recovery can be found here:
- High Availability and Disaster Recovery for DesktopNow SQL Server Databases

If you have configured SQL AlwaysOn with multi-subnet failover availability groups, you must handle the MultiSubnetFailover value in the database connection string for the relevant server. This can only be done using Server Configuration Platform Powershell cmdlets.
Note
For more information on configuring the database connection string, see Setting Up a New Server and Database in the AppSense Server Configuration Portal Scripting Guide.
Patch Installation

In this Section:

- **Patching** on page 95
- **Install and Uninstall a Patch Using the Management Center** on page 96
- **Install and Uninstall a Patch Using the Command Line** on page 98
- **Roll Back Service Packs** on page 99
Patching

AppSense DesktopNow products can be patched using a Windows Installer patch. A patch is an MSP file that, when installed, updates files and registry keys on an existing installed product. Installing an MSP can reduce system downtime because reboots are not always required. DesktopNow product patching gives all of the usual benefits associated with Windows Installer Patching, including ease of deployment and the ability to roll back to an earlier version. AppSense patches include the following:

- **Public Hotfix** - Issued publicly on myAppSense to address a widely reported issue and should only be installed to address the specific problem. Public Hotfixes are cumulative in that they contain all previous hotfixes. Public Hotfixes are distributed as an MSP.

- **Service Pack** - Contains all of the fixes from the last Private or Public Hotfix and any Service Packs, plus any fixes that have been found for which a Private or Public Hotfix was not issued. Service Packs are cumulative in that they contain all previous Service Packs. Service Packs are distributed as an MSP.

If a Service Pack is part of the product release media, the installer automatically installs them. Service Packs can also be installed or deployed using the same technology and techniques used when installing MSIs. Both Microsoft System Center and the AppSense Management Center can deploy MSPs. If neither of these products are available, service packs can be installed using the command line interface.

**Installation Order and Dependencies**

It is recommended that all components of a service pack are installed and that the PersonalizationServerXX.MSP is installed first. All other components have no required install order.

To view previously installed patches, navigate to Control Panel > Programs > Programs and Features > Installed Updates.

You have the following options to install a patch:

- Install and Uninstall a Patch Using the Management Center
- Install and Uninstall a Patch Using the Command Line
Install and Uninstall a Patch Using the Management Center

To install a patch using AppSense Management Center you must first upload the MSP and then assign it to a deployment group for deployment to the endpoints.

Upload an MSP

1. Open the Management Console and in the navigation pane select **Package Library**.
2. Click the required product, for example Environment Manager. The package library for the selected product displays in the work area.
3. From the Actions panel, select **Add Package**. The Browse for Package dialog displays.
4. Locate the required MSP file and click **Open**.

```
Note
If uploading a Service Pack the base MSI package must have been uploaded previously.
If uploading a Hotfix the base MSI package must have been uploaded. Additionally, if the Hotfix is applicable to a Service Pack, the Service Pack must also have been uploaded.
```

The Package Upload wizard displays.
5. Check the details of the selected package and optionally enter a description.
6. Click **Next** to start the upload.
7. When the upload has competed successfully, click **Finish**. The MSP can now be seen in the package library.

Deploy an MSP to an endpoint

1. In the Management Console, click **Home**. In the navigation pane, expand **Deployment Groups**, and then expand the group you want to deploy to and select **Packages**. The Packages work area displays a list of DesktopNow products and the assigned packages.
2. Highlight the required product package and from the Actions panel select **Change Agent Version**. A dialog to change the packages used by the group displays.
3. Select the required patch package. For example, 8.4 SP2 HF3

```
Note
The Management Center ensures that any dependencies for the selected patch are deployed to the endpoint.
Deployment Agent (CCA) 8.6 may be required to support deployment for the selected patch, if this is the case a warning message displays at the top of the work area.
```
4. Click **Finish**.
   The patch can now be seen in the Assigned Packages list.

5. Once all changes have been made, from the bottom of the work area, click **Review and Submit**.
   The Submit Changes For [Group Name] dialog displays.

6. Review the changes to be made to the deployment groups and click **Submit**.

   **Note**
   A warning may display informing you that changes to an agent can cause reboots at the times defined by the installation schedule.

7. If the warning message displays, click **Yes** to assign the changes. Alternatively, click **Submit** to assign the changes.
   The patch is deployed in accordance to the Deployment Group Installation Schedule.

### Uninstall an MSP

To uninstall a Service Pack or Hotfix using the Management Center:

1. In the Management Console, click **Home**. In the navigation pane, expand **Deployment Groups**, and then expand the group you want to deploy to and select **Packages**.
   The Packages work area displays a list of all the AppSense products and their associated packages.

2. Highlight the required Service Pack or Hotfix and click **Change Agent Version** from the Actions menu.
   The Change the packages used by this group dialog displays.

3. Select an alternative version and click **Finish**.

4. On the Assigned Packages work area click **Review and Submit**.
   The Submit Changes dialog displays.

5. Check the details are correct and click **Submit**.
   The uninstallation takes place in-line with the Deployment Group Installation Schedule.
Install and Uninstall a Patch Using the Command Line

AppSense DesktopNow patches can be installed from the command line as well as from the AppSense Management Center.

**Note**
It is recommended that logging is switched on when using the following commands. To enable logging add `/l*vx Patch.log` immediately after the `/i` or `/p`.

For example: `msiexec.exe /i Agent.msi /l*vx Patch.log`

**Command Line Install**

- To install or upgrade an MSI:
  
  `msiexec.exe /i Agent.msi`

- To silently install or upgrade an MSI:
  
  `msiexec.exe /i Agent.msi /qn`

- To install an MSP
  
  `msiexec.exe /p Agent.msp`

- To install an MSI and MSP in a single operation:
  
  `msiexec.exe /i Agent.msi PATCH=C:\FullPath\Agent.msp`

**Example:**

`msiexec.exe /p EnvironmentManagerAgent64.msp`

installs any files that have been amended as part of the patch for just Environment Manager 64-bit agent.

The following command installs the base version of the Environment Manager Agent (MSI) and the Environment Manager patch file (MSP) simultaneously:

`msiexec.exe /i EnvironmentManagerAgent64.msi PATCH=c:\fullpath\EnvironmentManagerAgent64.msp`

**Note**
A base version must be installed before the patch file can be applied.

If the patch file contains driver or hook files that are currently in use on the machine the patch is being applied to, you are informed that a reboot is required. If you chose to continue, the system is restarted when the patch has been applied.
Roll Back Service Packs

There are two ways to roll back, or uninstall, AppSense Service Packs:

- Using the Windows Control Panel
- Using AppSense Management Center

**Note**

If a service pack is uninstalled, the installation reverts to the previous latest build, whether a service pack or base version.

All agent and console service pack components except the Personalization Server component patch file (PersonalizationServerXX.msp) can be uninstalled.

Rolling Back Service Packs Using Windows Control Panel

The procedure used to roll back service packs varies depending on the Operating System:

**For Windows 7**

1. Navigate to **Control Panel > Programs > Programs and Features > Installed Updates**.
2. Highlight the selected patch and click **Uninstall**.
Rolling Back Service Packs Using Management Center

1. In the Management Center console, select Home.
2. Expand Deployment Groups, highlight the Deployment Group and select Packages.
   The Assigned Packages work area displays a list of all the AppSense products and their associated packages.
3. Highlight the required Service Pack or Hotfix and click Change Agent Version from the Actions menu.
   The Change the packages used by this group dialog displays.
4. Select an alternative version and click Finish.
5. On the Assigned Packages work area click Review and Submit.
   The Submit Changes dialog displays.
6. Check the details are correct and click Submit.
   The uninstallation takes place in accordance with the Deployment Group Installation Schedule.
Product Upgrades

In this Section:

- [Preparing to Upgrade](#) on page 102
- [Upgrade with the DesktopNow Installer](#) on page 102
- [Upgrade Application Manager](#) on page 106
- [Upgrade Environment Manager](#) on page 108
- [Upgrade Configurations](#) on page 110
- [Upgrade Servers](#) on page 110
Preparing to Upgrade

Existing AppSense software packages upgrade automatically during the installation process, including database schemas, agents, and configurations. Prior to an upgrade, it is recommended that you do the following:

- Back up all databases
- Save all product configuration packages from the console as MSI files.
- If necessary, save all earlier versions of the product agent software that you would like to maintain.
- You must upgrade the Management Server before you upgrade the Deployment Agent (CCA). The Management Server must be of the same version number or later than the installed Deployment Agent (CCA).
- Disconnect all users from the Personalization Servers.
- Take all Personalization Servers offline until the upgrade is complete.
- Run the whole suite installer to upgrade components.

Upgrade with the DesktopNow Installer

1. Run the Installer by executing the setup.exe file in the installation media. The Welcome screen displays.
2. Select **Upgrade** and click **Next**.
   The Product Selection screen displays.

   ![Image](image.png)

   **Note**
   Only the products currently installed display for you to select to upgrade.
   If you have server products installed, each instance is listed. Select the ones to upgrade.

3. Select the product components that you want to upgrade and click **Next**.
   ![Image](image.png)
   The Summary screen displays listing the products to be upgraded.
4. Click **Next**.

   The next step depends on whether all prerequisites are installed:
   - If any prerequisites are missing the Prerequisites screen displays.
     
     ![Prerequisites Screen](image)
     
     Select **Install All** to install all missing prerequisites.
     
     Once all components are installed the upgrade process starts immediately.
     
     - If there are no missing prerequisites the upgrade process starts.
   
5. If there are missing server prerequisites the Server Prerequisites screen displays, select **Install All** to install all missing components.
6. On completion the Upgrade Complete screen displays.

7. Click **Finish** to exit the Installer and open the User Assistance Portal.
   
   If you want to exit without opening the User Assistance Portal, deselect **Open User Assistance Portal on finish** and click **Finish**.
Upgrade Application Manager

If you are upgrading configurations used in previous versions of Application Manager, the introduction of the Process Rules and Group Management functionality may render the following parts of the configuration redundant:

- Trusted Applications
- Signature Groups
- Network Connection Groups
- URL Redirection and Custom Rules

Upgrades and Process Rules

If the Application Manager configuration contains Trusted Application rules, the upgrade will preserve the Trusted Applications feature's behavior although some functionality regarding the three Trusted Applications options may be lost.

The table below shows how the various Trusted Application states will be converted to Process rules during a configuration upgrade.

<table>
<thead>
<tr>
<th>Trusted Application State</th>
<th>Process Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>No Process rules added.</td>
</tr>
<tr>
<td>Disable Trusted Applications Checking</td>
<td>No Process rules added.</td>
</tr>
<tr>
<td>• Only when blocked by Trusted Ownership</td>
<td>For each Trusted Application defined:</td>
</tr>
<tr>
<td>• Always</td>
<td>- A new Process rule is created with the name Upgraded Trusted Application</td>
</tr>
<tr>
<td></td>
<td>Rule (*). Where * represents a number automatically incremented from 1 to the</td>
</tr>
<tr>
<td></td>
<td>number of Trusted Application rules present in the configuration being upgraded.</td>
</tr>
<tr>
<td></td>
<td>- A new Process Identifier is added to the newly created Process rule.</td>
</tr>
<tr>
<td></td>
<td>- If the Trusted Application rule was defined using a full file path then the</td>
</tr>
<tr>
<td></td>
<td>process identifier list has one file name entry with the exact same text.</td>
</tr>
<tr>
<td></td>
<td>- If the Trusted Application rule was defined using a digital signature then</td>
</tr>
<tr>
<td></td>
<td>the process identifier has one digital signature entry with the same digital</td>
</tr>
<tr>
<td></td>
<td>signature. Any file name information is preserved.</td>
</tr>
<tr>
<td></td>
<td>- For each of the trusted content entries for the Trusted Application rule, a</td>
</tr>
<tr>
<td></td>
<td>new Allowed Item is added. The Trusted Ownership setting is set to Off, for</td>
</tr>
<tr>
<td></td>
<td>all added entries.</td>
</tr>
</tbody>
</table>
Upgrades and Group Management

If the Application Manager configuration contains Signature Groups and Network Connection Groups, the upgrade directly converts them to Group Management and renames them Groups. The name of the Signature or Network Connection Group remains the same and the contents of the Signature or Network Connection Group remain the same.

To avoid any problems that may be encountered if the upgrade produces any duplicate names, each upgraded Group will be suffixed with its origin and that it was an upgrade.

Example

- A Signature Group called A, becomes a Group called A - Upgraded Signature Group.
- A Network Connection Group called B, becomes a Group called B - Upgraded Network Connection Group.

URL Redirection and Custom Rules

Custom rules and URL Redirection in Application Manager 10.0 differ considerably from versions 8.8 and 8.9. You can upgrade version 8.8 and 8.9 configurations by opening them in a version 10.0 console and saving them. This changes the configurations as follows:

- Custom rules are recreated using the new version 10.0 conditions, matching the behavior of the earlier version rules.
- URL Redirections are converted to Custom rules that contain:
  - Matching conditions for connection types, IP addresses, and port numbers.
  - Browser Control items for the sensitive URLs.

If you don't upgrade the configuration, the Application Manager Agent version 10.0 still reads the configuration, but the URL Redirection and Custom rules are ignored. The rest of the application still applies.
Upgrade Environment Manager

Environment Manager components must be upgraded in the following order:

<table>
<thead>
<tr>
<th>Order</th>
<th>Components</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All Personalization Servers and Databases</td>
<td>All Personalization Servers and Databases must be upgraded together - if you are using SQL replication then see separate best practice guides on upgrading the database and server.</td>
</tr>
<tr>
<td>2</td>
<td>All Consoles</td>
<td>Personalization Server is only compatible with the matching console version, so the consoles will need to be upgraded immediately after the servers. Note that the Policy configuration should not be upgraded until step 4.</td>
</tr>
<tr>
<td>3</td>
<td>All Agents</td>
<td>The Personalization components in the Environment Manager Agent are compatible with all Personalization Server versions. If you have configurations created with older consoles then you might need to upgrade agents and configurations simultaneously, one group at a time.</td>
</tr>
<tr>
<td>4</td>
<td>Configuration Files</td>
<td>When new agents have been deployed to all endpoints and are working successfully then any legacy policy configurations in use can be upgraded by the latest console and deployed.</td>
</tr>
</tbody>
</table>

Endpoint Configuration Merging

If you are using the Endpoint Configuration Merging function in Environment Manager, all configurations must be of the same EM product version. Upgrade all configurations before merging. For more information see the Environment Manager Help > configuration Endpoint Merging.

Conversion Rules

- Personalization Groups that used Global Desktop Settings that were Shared are placed in a **GlobalShared** Windows Settings Group.
- Personalization Groups that used Separate Global Desktop Settings are placed in the Windows Settings Groups to match the appropriate operating systems:
  - GlobalXP
  - GlobalVista
  - GlobalWin7
  - GlobalWin8
- Personalization Groups that used specific shared and separate Desktop Settings are placed in either of the following Windows Settings Groups:
  - [Personalization Group Name]_Shared
  - [Personalization Group Name]_OS
Session Data is placed in a **SessionData** Windows Settings Group.

Certificates and Credentials are placed in the **Security** Windows Settings Group.

### Note
These groups are created to ensure backward compatibility during the upgrade process. They are applied to Personalization Groups that are using Desktop Settings when the Environment Manager agent is upgraded to 8.6. It is recommended that they are replaced using the default 8.6 Windows Settings Groups wherever possible.

### Upgrading The Logon Trigger

In Environment Manager 8.5, a new Logon trigger structure was introduced replacing the single Logon trigger with three sub-triggers. This increases efficiency and speeds up login times, as Environment Manager actions can be configured to run at their most appropriate point during the user logon process:

- **Pre-Session** - Actions take effect before terminal services is notified of the logon. Registry, Group Policy, and Environment actions are compatible with this sub-trigger. During the upgrade, actions that were previously in the Logon Environment tab are moved here.

- **Pre-Desktop** - Actions take effect when the user logs on to the system but before the desktop shell has started. During the upgrade, actions that were previously in the Logon trigger are moved here.

- **Desktop Created** - Actions take effect after the desktop shell and Explorer has started. To improve efficiency and logon times, any non-critical Logon actions should be added to this trigger, for example, mapping drives and printers.

After upgrading the configuration:

- Logon Condition 1 has been moved from the Logon Environment tab to the Pre-Session trigger
- Nodes 1, 2, 3 and 4 have been moved from the Logon node to the Pre-Desktop trigger
- The Desktop Created sub-trigger has been added

If you do not upgrade the Logon trigger, the upgraded configuration will open with the single Logon trigger. You will be prompted to upgrade each time you open the configuration.

### Tip
The Logon trigger method can be changed at any time using the Advanced Settings in the Environment Manager console. See **Configuration Settings** in the **AppSense Environment Manager Policy Help** for further details.

### Upgrading Certificates

The client maintains both the legacy format and the saved certificates and credentials in the same profile. When a new client logs on with an old profile, the client uses the old format to restore the certificates if there is no new-format data, and saves it in the new format. From then on there is no roaming between legacy and new clients.
Personalization Operations Bulk Operations User Selection

The Personalization Operations utility, introduced in Environment 10.0, is a web-based console for managing personalization data. You can create bulk data operations for multiple users that apply to Personalization Groups, Active Directory (AD) groups, or specific users. The Active Directory (AD) group information in the database comes from the endpoints. When a user logs on using a version 10.0 endpoint, the endpoint provides a list of the AD groups to which the user belongs. Endpoints running earlier versions of Environment Manager do not provide this information, so in a newly-upgraded system there is no AD group information at all, and selecting users by AD group is not possible. As the endpoints are upgraded to version 10.0 and users log on, the database receives AD group information about the users and searching for AD groups works.

Upgrade Configurations

AppSense product configurations must be upgraded sequentially by major product version. To upgrade a configuration, open it in the latest product console. The console detects that the configuration was created in an earlier version and prompts you to upgrade. When the configuration is subsequently saved, it is saved as the latest version and it is ready to be deployed using a deployment mechanism.

Environment Manager policy configuration files are upgraded when opened in a combined or policy console. When you open a configuration created in a more recent version of the console, you are asked if you want to upgrade the configuration. Click Yes to upgrade. Clicking No opens the console with an empty configuration.

Save the configuration to complete the upgrade and ensure compatibility with the latest version of the agent, server, and console. Once the configuration has been saved it is ready to be deployed.

Note
Policy configurations cannot be upgraded in the Personalization only console. They can only be upgraded if opened in the policy or combined console.

Caution
Any work in progress configurations are deleted during the upgrade process, so they must be saved before upgrading.

Upgrade Servers

All servers connected to a database must be upgraded at the same time, for further information on upgrading servers refer to the Server Configuration on page 66.
Uninstall

In this Section:
- Uninstallation on page 112
Uninstallation

AppSense Environment Manager is uninstalled using the AppSense DesktopNow Installer.

1. Run the AppSense DesktopNow Installer by executing `setup.exe` on the installation media.

2. In the Welcome screen, select **Uninstall** and click **Next**.
3. The Summary screen lists the product selected to uninstall, and whether a reboot will be required.

4. Click **Uninstall** to start the uninstallation process.

5. The Product Uninstall screen displays, the progress of the uninstallation can be seen at the bottom of the screen.
Once all products have been uninstalled the Uninstall Complete screen displays.

6. Click **Finish** to close.

Any user created configurations will not be uninstalled with the product, the A*MP files must be manually deleted from where they were saved (default location: C:\ProgramData\AppSense\<product>).

The Deployment Agent (CCA) and any Packages will not automatically be removed from the managed endpoints.

If Microsoft SQL Server Express was installed during the installation process, this will not be uninstalled as part of the uninstallation process.

**Note**

If AppSense DesktopNow is installed in Evaluation mode then you must use **Control Panel > Add\Remove Programs** applet to uninstall.

If using this method ensure you uninstall each component of DesktopNow.
Appendices

In these Appendices:
- Multi Instance Command Line Installer on page 116
- Management Center MSI Custom Actions Usage on page 119
- Management Center Third Party Public Symbols Usage on page 138
Multi Instance Command Line Installer

In this Appendix:

- **The Command Line Installer** on page 117
- **Using InstallerCmd.exe** on page 117
- **InstallerCmd.exe Examples** on page 118
The Command Line Installer

The AppSense multi instance command line installer, InstallerCmd.exe, allows you to script installation of named instances.

The InstallerCmd.exe can be run from the Bin folder on the DesktopNow release media.

Using InstallerCmd.exe

The AppSense instance installer can be used to do the following actions:

- Install product server instances by name
- Patch specific product server instances
- Uninstall specific product server instances
- Display a list of available and installed product server instances

**Caution**

The command line generated is specific to the machine it is run on. It **must not** be transported to another machine.

**Install Options**

```
/i <Product.msi> <Instance Name> [Optional Parameter]
```

Launches the Windows Installer to install an instance with the specified name. Additional parameters understood by the Windows Installer can be passed at the end of this command.

```
/is <Product.msi> <Instance Name> [Optional Parameter]
```

Displays (but does not run) the Windows Installer command required to install an instance with the specified name. Additional parameters understood by the Windows Installer can be passed at the end of this command.

**Patch Options**

```
/p <Patch.msp> <Instance Name> [Optional Parameter]
```

Launches the Windows Installer to patch an instance with the specified name. Additional parameters understood by the Windows Installer can be passed at the end of this command.

```
/ps <Patch.msp> <Instance Name> [Optional Parameter]
```

Displays (but does not run) the Windows Installer command required to patch an instance with the specified name. Additional parameters understood by the Windows Installer can be passed at the end of this command.
Uninstall Options
/x <Product.msi> <Instance Name> [Optional Parameter]
Launches the Windows Installer to uninstall an instance with the specified name. Additional parameters understood by the Windows Installer can be passed at the end of this command.

/xs <Product.msi> <Instance Name> [Optional Parameter]
Displays (but does not run) the Windows Installer command required to uninstall an instance with the specified name. Additional parameters understood by the Windows Installer can be passed at the end of this command.

Display Options
/e <Product.msi>
Lists available and installed instances associated with the specified product. The instance name is displayed for installed instances.

Tip
Details of Optional Parameters can be found in the help for Windows Installer.
Run:
MSIEXEC /?

InstallerCmd.exe Examples

The following are examples of how to use AppSense InstallerCmd.exe:

Install Example
InstallerCmd.exe /i <path>\ManagementServer64.msi TestInstance /q
Installs an instance, named TestInstance, of the AppSense Management Server and passes the /q parameter to the Windows Installer resulting in no user interface being displayed during installation.

Uninstall Example
InstallerCmd.exe /x <path>\PersonalizationServer64.msi TestInstance /l* uninstall.log
Uninstalls the AppSense Personalization Server instance named TestInstance and saves the log to uninstall.log
Management Center MSI Custom Actions Usage

In this Appendix

- MSI Custom Actions on page 120
- Management Console Custom Actions on page 120
- Management Server Custom Actions on page 127
- Client Communications Agent Custom Actions on page 134
### MSI Custom Actions

There are a number of Custom Actions used within the Management Center that provide enhanced capabilities to the standard action. Each Custom Action relates to one of the following MSI’s used in the installation of the Management center:

- ManagementConsole32.msi or ManagementConsole64.msi
- ManagementServer64.msi
- ClientCommunicationAgent32.msi or ClientCommunicationAgent64.msi

The custom actions are listed in the following tables:

#### Management Console Custom Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>doGetManagementServerURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1 – Assembly: InstallerActions, Method: GetManagementServerURL</td>
</tr>
<tr>
<td>Description</td>
<td>Reads the last used server from ServerSettings.xml and sets the MANAGEMENTSERVERURL msi property to the most recently used server. This is run during a GUI install to pre-populate the management server dialog with a default management server when not upgrading an old version. Reads: [%APPDATA%\AppSense\ServerSettings.xml]</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>dStripServerPath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1 – Assembly: InstallerActions, Method: StripServerPath</td>
</tr>
<tr>
<td>Description</td>
<td>Removes the &quot;ManagementServer&quot; path from the MSI property MANAGEMENTSERVERURL. Reads property MANAGEMENTSERVERURL. Sets property MANAGEMENTSERVERURL.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>EnterServerWarning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>4134 – Embedded VBScript</td>
</tr>
<tr>
<td>Description</td>
<td>Displays a message box stating “Please enter the Management Server URL” if the user forgets.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>N/A</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>CheckURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1 – Assembly: InstallerActions, Method: TestURL</td>
</tr>
<tr>
<td>Description</td>
<td>Checks whether a given URL is well-formed. Reads property MANAGEMENTSERVERURL. Sets MANAGEMENTSERVERURL to canonical version if URL is good. Sets URL_OK to either “1” if good or empty string if bad URL.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
</tbody>
</table>
### MSI Custom Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>BadURL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>4134 – Embedded VBScript</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Displays a message box saying &quot;URL is invalid&quot; if the user types a bad URL. Run if property URL_OK is empty.</td>
</tr>
<tr>
<td><strong>Occurs During</strong></td>
<td>Install</td>
</tr>
<tr>
<td><strong>Changes System State</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Reversible</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Immediate</td>
</tr>
<tr>
<td><strong>Has Rollback</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>MSIProcessMessage</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Mis-uses MSI Prefixes</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Set Registry Keys</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>GacUtil</strong></td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>ExecuteManagementConsoleUninstall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>210 – executable: Console.exe</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Invokes the console executable in silent mode to perform uninstall tasks. Deletes: [%LOCALAPPDATA%\AppSense\ManagementConsole]</td>
</tr>
<tr>
<td><strong>Occurs During</strong></td>
<td>Uninstall</td>
</tr>
<tr>
<td><strong>Changes System State</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Reversible</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>In a script</td>
</tr>
<tr>
<td><strong>Has Rollback</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>MSIProcessMessage</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Mis-uses MSI Prefixes</strong></td>
<td>No</td>
</tr>
</tbody>
</table>
### AddServer

<table>
<thead>
<tr>
<th>Name</th>
<th>AddServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1025 – assembly: InstallerActions, Method: AddServer</td>
</tr>
<tr>
<td>Description</td>
<td>Adds the value in [MANAGEMENTSERVERURL] into the list of known management servers in ManagementServers.xml, creating it if needed. If a new entry is added into this file, then it is marked as new with a special tag. This tag is used to remove the entry on rollback, and removed during the commit phase.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>deferred</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>Yes</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

### RemoveServer

<table>
<thead>
<tr>
<th>Name</th>
<th>RemoveServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1281 – assembly: InstallerActions, Method: RemoveServer</td>
</tr>
<tr>
<td>Description</td>
<td>Removes [MANAGEMENTSERVERURL] from the ManagementServers.xml file if it was marked as new</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>rollback</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>RemoveMarkerTags</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>1537 –assembly: InstallerActions, Method: RemoveMarkerTags</td>
</tr>
<tr>
<td>Description</td>
<td>Removes the 'new' tags from the ManagementServers.xml file</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>commit</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>
### MSI Custom Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>TurnOffGeneratePublisherEvidence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>1025 – assembly: InstallerActions, method TurnOffGeneratePublisherEvidence</td>
</tr>
</tbody>
</table>
| **Description**          | This action reads the ManagementConsole.exe.config file and ensures that the generatePublisherEvidence element exists and is set to false. This will not have any effect during a normal install as the installed file has this set. Writes to:  
  - [CONSOLEINSTALLDIR]ManagementConsole.exe.config  
  - [CONSOLEINSTALLDIR]ManagementConsole.exe.config.tmp |
| **Occurs During**        | Install |
| **Changes System State** | Yes |
| **Reversible**           | N/A |
| **Execute**              | deferred |
| **Has Rollback**         | Yes |
| **MSIProcessMessage**    | Yes |
| **Mis-uses MSI Prefixes**| No |
| **Set Registry Keys**    | No |
| **GacUtil**              | No |

<table>
<thead>
<tr>
<th>Name</th>
<th>TurnOffGeneratePublisherEvidenceCommit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>1025 – assembly: InstallerActions, method TurnOffGeneratePublisherEvidenceCommit</td>
</tr>
</tbody>
</table>
| **Description**          | This action deletes any tmp files created by the TurnOffGeneratePublisherEvidence action  
  - Deletes: [CONSOLEINSTALLDIR]ManagementConsole.exe.config.tmp |
<p>| <strong>Occurs During</strong>        | Install |
| <strong>Changes System State</strong> | Yes |
| <strong>Reversible</strong>           | N/A |
| <strong>Execute</strong>              | commit |
| <strong>Has Rollback</strong>         | N/A |
| <strong>MSIProcessMessage</strong>    | Yes |
| <strong>Mis-uses MSI Prefixes</strong>| No |
| <strong>Set Registry Keys</strong>    | No |
| <strong>GacUtil</strong>              | No |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>TurnOffGeneratePublisherEvidenceRollback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1025 – assembly: InstallerActions, method TurnOffGeneratePublisherEvidenceRollback</td>
</tr>
<tr>
<td>Description</td>
<td>This action copies any tmp files made by the TurnOffGeneratePublisherEvidence action over the original files, then deletes the tmp files. Writes To: [CONSOLEINSTALLDIR]ManagementConsole.exe.config. Deletes: [CONSOLEINSTALLDIR]ManagementConsole.exe.config.tmp</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Rollback</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>WixFailWhenDeferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1025 – assembly: WixCA, method WixFailWhenDeferred</td>
</tr>
<tr>
<td>Description</td>
<td>Part of the WiX library. <a href="http://wix.sourceforge.net/manual-wix3/wixfailwhendeferred.htm">http://wix.sourceforge.net/manual-wix3/wixfailwhendeferred.htm</a> This action simply fails the install during the deferred execution stage. Set the property WIXFAILWHENDEFERRED=1 to cause the failure.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Deferred</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>
### MSI Custom Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>FolderToDelete</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>This action deletes the specified directory</td>
</tr>
<tr>
<td><strong>Occurs During</strong></td>
<td>Uninstall</td>
</tr>
<tr>
<td><strong>Changes System State</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Reversible</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Deferred</td>
</tr>
<tr>
<td><strong>Has Rollback</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>MSIProcessMessage</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Mis-uses MSI Prefixes</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Set Registry Keys</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>GacUtil</strong></td>
<td>No</td>
</tr>
</tbody>
</table>

### Management Server Custom Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>WixUIValidatePath</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Part of the Wix library.</td>
</tr>
<tr>
<td><strong>Occurs During</strong></td>
<td>Install</td>
</tr>
<tr>
<td><strong>Changes System State</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Reversible</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td>Immediate</td>
</tr>
<tr>
<td><strong>Has Rollback</strong></td>
<td>N/A</td>
</tr>
<tr>
<td><strong>MSIProcessMessage</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Mis-uses MSI Prefixes</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Set Registry Keys</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>GacUtil</strong></td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>WixUIPrintEula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Part of the Wix library.</td>
</tr>
<tr>
<td>Name</td>
<td>GetPlatform</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Type</td>
<td>1 –assembly: InstallerActions, Method: GetPlatform</td>
</tr>
<tr>
<td>Description</td>
<td>Sets PLATFORM with either “32-bit” or “64-bit” depending on the platform being installed on.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>N/A</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>TurnOffGeneratePublisherEvidence1 &amp; 2</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>1025 –assembly: InstallerActions, Method: TurnOffGeneratePublisherEvidence</td>
</tr>
</tbody>
</table>
| Description | This action reads the ManagementConsole.exe.config file and ensures that the generatePublisherEvidence element exists and is set to false. This will not have any effect during a normal install as the installed file has this set. This task is split into 2 due to limitations in the amount of information that can be passed in a single windows installer property. Writes to:  
  - [BINDIR]AlertsServices.exe.config  
  - [BINDIR]DeploymentService.exe.config  
  - [BINDIR]EventsDispatcher.exe.config  
  - [BINDIR]SchedulerService.exe.config  
  - [BINDIR]ServerConfiguration.exe.config  
  - [BINDIR]AlertsServices.exe.config.tmp  
  - [BINDIR]DeploymentService.exe.config.tmp  
  - [BINDIR]EventsDispatcher.exe.config.tmp  
  - [BINDIR]SchedulerService.exe.config.tmp  
  - [BINDIR]ServerConfiguration.exe.config.tmp |
| Occurs During | Install |
| Changes System State | Yes |
| Reversible | N/A |
| Execute | Deferred |
| Has Rollback | Yes |
| MSIProcessMessage | Yes |
| Mis-uses MSI Prefixes | No |
| Set Registry Keys | No |
| GacUtil | No |

<table>
<thead>
<tr>
<th>Name</th>
<th>TurnOffGeneratePublisherEvidenceCommit1 &amp; 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1025 –assembly: InstallerActions, Method: TurnOffGeneratePublisherEvidence-Commit</td>
</tr>
</tbody>
</table>
| Description | This action deletes any .tmp files created by TurnOffGeneratePublisherEvidence1 & 2. Deletes:  
  - [BINDIR]AlertsServices.exe.config.tmp  
  - [BINDIR]DeploymentService.exe.config.tmp  
  - [BINDIR]EventsDispatcher.exe.config.tmp  
  - [BINDIR]SchedulerService.exe.config.tmp  
  - [BINDIR]ServerConfiguration.exe.config.tmp |
<p>| Occurs During | Install |
| Changes System State | Yes |
| Reversible | N/A |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>LaunchServerUninstall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>3154 – executable: ServerConfig.exe</td>
</tr>
</tbody>
</table>
| Description               | This action launches the SCU with the /uninstall flag to perform generic uninstall actions:  
  - Stops all includes services  
  - Unregisters all services  
  - Stops all IIS AppPools  
  - Deletes all web directories & website filesystem directories  
  - Removes all AppPools |
| Occurs During             | Uninstall             |
| Changes System State      | Yes                   |
| Reversible                | N/A                   |

<table>
<thead>
<tr>
<th>Name</th>
<th>GacUtil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute</td>
<td>Deferred</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>Yes</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>LaunchServerBeginUpgrade</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>3154 – executable: ServerConfig.exe</td>
</tr>
<tr>
<td>Description</td>
<td>This action launches the SCU with the /beginupgrade flag to perform generic upgrade actions: Stops all installed services</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Uninstall</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Deferred</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>No</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>LaunchServerEndUpgrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>3154 – executable: ServerConfig.exe</td>
</tr>
</tbody>
</table>
| Description              | This action launches the SCU with the /endupgrade flag to perform generic upgrade actions: Writes to the following webconfigs:  
  - [BINDIR]\ActiveProductDefinition\ManagementServer.xml  
  - [BINDIR]\AlertsServices.exe.config  
  - [BINDIR]\DeploymentService.exe.config  
  - [BINDIR]\EventsDispatcher.exe.config  
  - [BINDIR]\SchedulerService.exe.config  
  - [BINDIR]\ServerConfiguration.exe.config |
<p>| Occurs During            | Install                                       |
| Changes System State     | Yes                                           |
| Reversible               | N/A                                           |
| Execute                  | Deferred                                      |
| Has Rollback             | No                                            |
| MSIProcessMessage        | Yes                                           |
| Mis-uses MSI Prefixes    | No                                            |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>RestartServicesPostRepair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1 - Assembly: InstallerActions, Method: RestartServicesPostRepair</td>
</tr>
<tr>
<td>Description</td>
<td>This action restarts the AppSense services at the end of a repair operation. NOTE: This custom action does not require a corresponding commit custom action.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Repair</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Deferred</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>Yes</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>SetAppSenseServicesToRestart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1 - Assembly: InstallerActions, Method: SetAppSenseServicesToRestart</td>
</tr>
<tr>
<td>Description</td>
<td>This action reads a CustomActionData property listing all the AppSense Windows Services and determines which are running and which are currently stopped. This information is recorded and used if a rollback occurs to ensure that on completion of a rollback the service system state is the same.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Repair</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>No</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
</tbody>
</table>
**Set Registry Keys** | No
---|---
**GacUtil** | No

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th>RestartServicesPostRepairRollBack</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>1 – Assembly: InstallerActions, Method: RestartServicesPostRepairRollBack</td>
</tr>
</tbody>
</table>
| **Description** | This action ensures that if a rollback occurs during a repair operation, the AppSense windows services have their initial service status as determined by the SetAppSenseServicesToRestart custom action.  
NOTE: this custom action does not require a corresponding commit custom action. |
| **Occurs During** | Repair |
| **Changes System State** | Yes |
| **Reversible** | N/A |
| **Execute** | Deferred |
| **Has Rollback** | Yes |
| **MSIProcessMessage** | Yes |
| **Mis-uses MSI Prefixes** | No |
| **Set Registry Keys** | No |
| **GacUtil** | No |
### Client Communications Agent Custom Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppendAppSenseInstallMgrCM</td>
<td>Reads the registry string at HKLM\System\CurrentControlSet\Control\Network-Provider\Order\ProviderOrder and then appends a comma separated string at the end which is the name of the AppSense credential manager hook dll.</td>
</tr>
<tr>
<td>RemoveAppSenseInstallMgrCM</td>
<td>Reads the registry string at HKLM\System\CurrentControlSet\Control\Network-Provider\Order\ProviderOrder and then removes the AppSense credential manager hook dll name from the string without corrupting the current network provider order.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 - Assembly: InstallerActions, Method: AppendAppSenseInstallMgrCM</td>
<td></td>
</tr>
<tr>
<td>Type 1 - Assembly: InstallerActions, Method: RemoveAppSenseInstallMgrCM</td>
<td></td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>Yes</td>
</tr>
<tr>
<td>Reversible</td>
<td>Yes</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>Yes</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>Yes</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
<tr>
<td>Execute</td>
<td>Deferred</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>No</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>Yes</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>
### EnterServerWarning

<table>
<thead>
<tr>
<th>Name</th>
<th>EnterServerWarning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>4134 – Embedded VBScript</td>
</tr>
<tr>
<td>Description</td>
<td>Displays a message box saying “Please enter the Management Server URL” if the user forgets during a manual install</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

### CheckURL

<table>
<thead>
<tr>
<th>Name</th>
<th>CheckURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>4134 – Embedded VBScript</td>
</tr>
<tr>
<td>Description</td>
<td>Checks whether a given URL is well-formed. Reads property MANAGEMENTSERVERURL. Sets MANAGEMENTSERVERURL to canonical version if URL is good. Set URL_OK to either “1” if good or empty string if bad URL</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>Yes</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>
Name | BadURL  
---|---  
Type | 4134 – Embedded VBScript  
Description | Displays a message box saying “URL is invalid” if the user types a bad URL. Run if property URL_OK is empty.  
Occurs During | Install  
Changes System State | No  
Reversible | N/A  
Execute | Immediate  
Has Rollback | N/A  
MSIProcessMessage | N/A  
Mis-uses MSI Prefixes | No  
Set Registry Keys | No  
GacUtil | No

Name | WixUIPrintEula  
---|---  
Type | 65 – 3rd party assembly: WixUIWixca, Method: PrintEula  
Description | Displays a message box saying “URL is invalid” if the user types a bad URL. Run if property URL_OK is empty.  
Occurs During | Install  
Changes System State | No  
Reversible | N/A  
Execute | Immediate  
Has Rollback | N/A  
MSIProcessMessage | N/A  
Mis-uses MSI Prefixes | No  
Set Registry Keys | No  
GacUtil | No

Name | CheckBITS  
---|---  
Type | 1 –assembly: InstallerActions, Method: CheckBITS
### MSI Custom Actions

#### FormatWebSiteURL

<table>
<thead>
<tr>
<th>Description</th>
<th>Checks if BITS is installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>N/A</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>FormatWebSiteURL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>1 –assembly: InstallerActions, Method: FormatWebSiteURL</td>
</tr>
<tr>
<td>Description</td>
<td>Appends a forward slash (/) to URL if none present and adds port number.</td>
</tr>
<tr>
<td>Occurs During</td>
<td>Install</td>
</tr>
<tr>
<td>Changes System State</td>
<td>No</td>
</tr>
<tr>
<td>Reversible</td>
<td>N/A</td>
</tr>
<tr>
<td>Execute</td>
<td>Immediate</td>
</tr>
<tr>
<td>Has Rollback</td>
<td>N/A</td>
</tr>
<tr>
<td>MSIProcessMessage</td>
<td>N/A</td>
</tr>
<tr>
<td>Mis-uses MSI Prefixes</td>
<td>No</td>
</tr>
<tr>
<td>Set Registry Keys</td>
<td>No</td>
</tr>
<tr>
<td>GacUtil</td>
<td>No</td>
</tr>
</tbody>
</table>
Management Center Third Party Public Symbols Usage

Public symbols are available for all AppSense Management Server binaries by contacting Support. The AppSense Management Server also uses a number of third-party binaries which may be available by contacting the vendor; these binaries are listed below.

- DevExpress_XtraCharts_<version>.dll
- DevExpress_Charts_<version>_Core.dll
- DevExpress_XtraReports_<version>.dll
- DevExpress_XtraGrid_<version>.dll
- DevExpress_XtraTreeList_<version>.dll
- DevExpress_Data_<version>.dll
- DevExpress_XtraBars_<version>.dll
- DevExpress_OfficeSkins_<version>.dll
- DevExpress_BonusSkins_<version>.dll
- DevExpress_XtraEditors_<version>.dll
- DevExpress_Utils_<version>.dll
- DevExpress_XtraNavBar_<version>.dll
- DevExpress_XtraLayout_<version>.dll
- DevExpress_Scheduler_<version>.dll
- DevExpress_XtraRichEdit_<version>.dll
- DevExpress_XtraPrinting_<version>.dll
- log4net.dll

Where <version> is the version number of the library.
Note
Public symbols for Microsoft binaries are available from
Glossary

- AAMP
- Adler-32
- ADM
- AEMP
- agent
- Allowed Items
- AM Web Service
- AMDX
- ANAC
- APMP
- App-V
- Application Group
- Application Limit
- Application Network Access Control (ANAC)
- Application Termination
- AppSense Application Manager Package (AAMP)
- AppSense Environment Manager Package (AEMP)
- AppSense Performance Manager Package (APMP)
- Audit Only
- base configuration
- cascadeBHO.dll
- Citrix XenApp
- client computer
- Client Debug Setup Utility
- COM
- configuration
- configuration file
- Configuration Profiler
- Constant Special Item ID List
- CSIDL
- DAC
- Database Management System (DBMS)
- DBMS
- Denied Items
- Deployment Agent (CCA)
- DFS
- digital signature
- Discretionary Access Control (DAC)
- Distributed File System (DFS)
- DLL
- DNS
- Domain Name System (DNS)
- Dynamic Link Library (DLL)
- EM
- EM Tools
- EmMon
- EMP
- EMP File Utility
- EMP Migrate Utility
- EMP Registry Utility
- Endpoint Analysis (EPA)
- Environment Manager (EM)
- Environment Manager Logging Setup Tool
- Environment Manager Monitor (EmMon)
- Environment Manager Personalization (EMP)
- Environment Manager Support console
- EPA
- FBR Explorer
- filter driver
- fixed node
- Folder Redirection
- Globally Unique Identifier (GUID)
- Group Management
- Group Policy
- GUID
- Local Security Authority (LSA)
- lockdown
- LSA
- managed application
- Management Server
- Masquerading Applications
- Microsoft Administrative Template (ADM)
- Microsoft Administrative Template XML-Based (ADMX)
- Microsoft Application Virtualization (App-V)
- MSI
- NetBIOS
- Network Basic Input/Output System (NetBIOS)
- Network Connection Item
- New Technology File System (NTFS)
- node
- NTFS
- ODBC
- Open Database Connectivity (ODBC)
- Open Software Description (OSD)
- organizational unit
- OSD
- OU
- Personalization Analysis
- Personalization Group
- Personalization Server
- Personalization Virtualization Component (PVC)
- Privilege Discovery
- Process Rules
- PVC
- registry hive
- regular expression
- Restricted
- reusable condition
- reusable node
- Secure Hash Identifier (SHA-1)
- Security Identifier SID
- Security Level
- Self Heal
- Self-Authorizing
- Server Configuration Portal (SCP)
- SHA-1
- SHA-256
- SID
- Site
- TCP
- Time Limits
- Transmission Control Protocol (TCP)
- trigger
- Trusted Applications
- Trusted Ownership
- Trusted Vendors
- UNC
- Universal Naming Convention (UNC)
- Unrestricted
- User Personalization
- User Privileges Management
- User Virtualization (UV)
- UV
- UV Service
- VDI
- Virtual Desktop Infrastructure (VDI)
- Windows Management Instrumentation (WMI)
- Windows Script Host (WSH)
- WMI
- WSH
AAMP
See AppSense Application Manager Package.

Adler-32
A checksum algorithm that can be applied to application files. See also: SHA-1 and SHA-256.

ADM
See Microsoft Administrative Template.

AEMP
See AppSense Environment Manager Package.

agent
An executable component that implements the product configuration settings. The DesktopNow agent is software that runs as a Windows service to carry out tasks on a computer, as specified by the configuration deployed to that computer.

Allowed Items
Files, folders, drives or digitally signed files or groups of files in an DesktopNow configuration Privilege Discovery which are allowed to run when file execution requests are matched with the rule security settings and would otherwise be denied by other configuration settings.
See also: Denied Items

AM Web Service
The Application Manager web service that can be installed on any machine and used to collect data from the enabled Privilege Discovery feature.

AMDX
See Microsoft Administrative Template XML-Based.

ANAC
See Application Network Access Control.

APMP
See AppSense Performance Manager Package.

App-V
See Microsoft Application Virtualization.

Application Group
Environment Manager - A collection of applications with common registry keys and folders that are managed as a single group for Environment Manager Personalization.
Performance Manager - A custom list of applications to which you assign rules for controlling CPU, memory or disk resources. Application Groups are associated with particular types of applications, desktop applications, services, or 16-bit applications.

Application Limit

An Application Manager setting, applied to an Allowed Item, that specifies the number of instances of an application that a user can run.

Application Network Access Control (ANAC)

An approach to computer network security that controls outbound network connections by IP Address, host name, URL, UNC, or port, based on the outcome of rules processing. In Application Manager ANAC is implemented using Network Connection Items and Reverse DNS Lookup.

Application Termination

An Application Manager feature that allows you to set triggers, behavior, and warning messages for terminating applications on managed computers.

AppSense Application Manager Package (AAMP)

The configuration file that stores settings created in Application Manager Configuration and is then deployed to endpoints. The agent uses the configuration settings to determine whether or not an execute request is to be denied.

AppSense Environment Manager Package (AEMP)

The configuration file that stores settings created in Environment Manager Policy Configuration and is then deployed to endpoints to manage the policy configuration on those endpoints as defined in the settings.

AppSense Performance Manager Package (APMP)

The configuration file that stores the policies created in Performance Manager for managing performance and resources on managed endpoints, and is then deployed to endpoints to manage performance as defined.

Audit Only

Security Level assigned to users, groups or devices in an Application Manager Privilege Discovery that audits events according to the auditing configuration without applying the rule. Used for passive monitoring in evaluations to assess application usage on the host environment.

base configuration

In endpoint configuration merging, the first configuration in the merge onto which the other configurations are added.
cascadeBHO.dll
An application Manager Browser Helper Object (BHO) loaded by Internet Explorer that is used as part of the URL Redirection and Elevated websites features.

Citrix XenApp
A thin client product that allows users to connect their corporate applications. XenApp can either host applications on central servers and allow users to interact with them remotely or stream and deliver them to user devices for local execution.

client computer
The device on which user login sessions are hosted.

Client Debug Setup Utility
An Environment Manager standalone utility, installed from the EM Tools Installer, which enables and disables the generation of diagnostic logs used by AppSense Support when diagnosing reported issues.

COM
A binary-interface standard for software components introduced by Microsoft. It is used to enable inter-process communication and dynamic object creation in programming languages.

configuration
A collection of settings created in the product console, in which a navigation tree of component settings graphically represents the configuration while it is created and modified by the administrator.

configuration file
A DesktopNow configuration saved as a native configuration file (.aamp, .aemp, .apmp) or an XML file, or exported from the product console in Windows Installer MSI file format. The file can be installed on any computer and the configuration rules applied when a DesktopNow Agent is present and running as a service on the computer.

See also: AppSense Application Manager Package (AAMP), AppSense Environment Manager Package (AEMP), and AppSense Performance Manager Package (APMP)

Configuration Profiler
In the product console, a feature that generates reports detailing the current settings in the configuration. Filtering options allow you to query settings affecting specific users or groups, devices and files or folders.

Constant Special Item ID List
A set of values that provide a unique, system-independent way to identify special solders used frequently by applications, but which may not have the same name or location on any given system.
**CSIDL**

See Constant Special Item ID List.

**DAC**

See Discretionary Access Control.

**Database Management System (DBMS)**

Computer management software that manages databases installed on a system or network.

**DBMS**

See Database Management System.

**Denied Items**

Items that are not allowed to run when file execution requests are matched with the rule security settings and would otherwise be allowed by other configuration settings. See also: Allowed Items

**Deployment Agent (CCA)**

A software service installed on computers to provide communication between the product agent running on a managed computer and the AppSense Management Center. The Deployment Agent (CCA) sends event data generated by the product agents to the Management Server and also polls the Management Server to manage the download and installation of software configuration, agent and package updates.

**DFS**

See Distributed File System.

**digital signature**

A means to accurately validate the authenticity of a file according to the actual contents of the file itself by using a mathematical technique (a cryptographic hashing algorithm). If the file is altered in any way then the SHA-1 hash is also altered. Application Manager has a Signature Wizard that allows you to apply digital signatures either to an individual file or a group. The signature can be used as a security measure when files as Allowed Items, Denied Items and Trusted Vendors. Application Manager uses the industry standard SHA-1, SHA-256 and Adler-32 hashes.

**Discretionary Access Control (DAC)**

A type of access control defined by the Trusted Computer System Evaluation Criteria as a means of restricting access to objects based in the identity of subjects and/or groups to which they belong.

**Distributed File System (DFS)**

Any file system that allows access to files from multiple hosts sharing via a computer network.
DLL
See Dynamic Link Library.

DNS
See Domain Name System.

Domain Name System (DNS)
A system that translates a computer’s fully qualified domain name into an IP address.

Dynamic Link Library (DLL)
A collection of software functionality that a running executable can call upon as needed, for example to communicate with a specific device such as a printer or to perform particular tasks.

EM
See Environment Manager.

EM Tools
A collection of standalone tools installed from the EM Tools Installer, which assist administrators when creating configurations and working with the Personalization Database.

EmMon
See Environment Manager Monitor.

EMP
See Environment Manager Personalization.

EMP File Utility
A standalone utility, installed from the EM Tools Installer, which imports and exports files to and from the Environment Manager Personalization database.

EMP Migrate Utility
A standalone utility, installed from the EM Tools Installer, which allows user data to be copied from a source database to a target database.

EMP Registry Utility
A standalone utility, installed from the EM Tools Installer, which relocates registry keys when multiple versions are stored in user profiles in the Personalization database.

Endpoint Analysis (EPA)
In Application Manager, the functionality to monitor managed endpoints to provide a list of applications that are present and that have run on a particular computer.
Environment Manager (EM)
   An AppSense user virtualization solution that ensures users always receive a consistent, predictable, and personalized working experience.

Environment Manager Logging Setup Tool
   In Environment Manager, a standalone utility, installed from the EM Tools Installer, which enables and disables the generation of diagnostic logs used by AppSense Support when diagnosing reported issues.

Environment Manager Monitor (EmMon)
   In Environment Manager, a standalone utility, installed from the EM Tools Installer, that provides a user interface to view and analyze log files from the Environment Manager Agent. This tool replaces Environment Manager Log Viewer.

Environment Manager Personalization (EMP)
   An Environment Manager feature that captures a users application and desktop changes to a central database and reapplies them for the user upon logon or application start, regardless of operating system or delivery mechanism.

Environment Manager Support console
   An abridged version of the Environment Manager console, available to users with the Support Console role, that offers read-only access to Environment Manager configurations and full access to Personalization Analysis functionality.

EPA
   See Endpoint Analysis.

FBR Explorer
   A standalone utility, installed from the EM Tools Installer, which open any File Based Registry (FBR) enabling the keys and values stored for an application to be viewed.

filter driver
   An optional driver that adds value to or modifies the behavior of a device. A filter driver can service one or more devices.

fixed node
   In the product console navigation tree, a node that cannot be deleted or edited.

Folder Redirection
   A Microsoft Windows component that can be controlled by group policy. Environment Manager simplifies its configuration.

Globally Unique Identifier (GUID)
   A unique reference number used as an identifier in computer software.
Group Management

In Application Manager, a library for compiling reusable groups of files, folders, drives, signatures, and network connections that can be associated with rules in the configuration. For example, Groups can be used to manage licenses for a suite of software or common sets of applications for assigning to certain user groups.

Group Policy

A set of configuration rules that can be defined to manage the working environment of users and computers.

GUID

See Globally Unique Identifier.

Local Security Authority (LSA)

A required Windows component that deals with login authentication and security policies.

lockdown

An Environment Manager mechanism to restrict or disable access to specific application and operating system functionality, keyboard shortcuts, Microsoft Office application menus, and toolbars.

LSA

See Local Security Authority.

managed application

Applications that have their settings and any changes made to those settings captured by Environment Manager Personalization.

Management Server

The machine on which product configurations and configuration versions are stored, from which configurations can be deployed to machines designated by the administrator.

Masquerading Applications

Applications that are allowed to run against the personalization caches of another application in order to use its Personalization data. This is especially useful when using multiple applications on multiple endpoints in situations where using a Personalization Group is not feasible.

Microsoft Administrative Template (ADM)

The file type of the administrative template files that are used by Environment Manager Group Policies to describe where registry-based policy settings are stored in the registry.
Microsoft Administrative Template XML-Based (ADMX)

The file type for XML based administrative template files that are used by Environment Manager Group Policies to describe where registry based policy settings are stored in the registry for Microsoft Windows Vista and Server 2008.

Microsoft Application Virtualization (App-V)

An application virtualization and application streaming solution that allows application to be deployed in real-time to clients and virtual application servers. It was formerly known as Microsoft SoftGrid.

MSI

The filetype for a Microsoft Installer file, an installer file package format used by Windows.

NetBIOS

See Network Basic Input/Output System.

Network Basic Input/Output System (NetBIOS)

A program that allows applications on different computers to communicate within a local area network (LAN).

Network Connection Item

In Application Manager, a network resource that can be added to a rule as an Allowed or Denied Item.

New Technology File System (NTFS)


node

Application Manager and Performance Manager - A branch in the navigation tree in the product console.

Environment Manager - In a configuration, a container that houses conditions and actions within triggers.

NTFS

See New Technology File System.

ODBC

See Open Database Connectivity.

Open Database Connectivity (ODBC)

A standard software interface for accessing Database Management Systems (DBMS), making communication between applications and databases easier.
Open Software Description (OSD)
A file format generated by App-V to define how an application is launched and configured.

Organizational unit
A container that holds users and computers in Active Directory.

OSD
See Open Software Description.

OU
See organizational unit.

Personalization Analysis
An Environment Manager feature that monitors which applications are being controlled by Environment Manager, including how much data is being stored, and enables managers to convert discovered applications to UV Services and to roll back to Personalization restore points.

Personalization Group
A group of users, based on common requirements, that can be treated as a single entity for Environment Manager Personalization configuration. This allows group members to have the same managed applications, Application Groups, and user personalization settings.

Personalization Server
The server that acts as a broker between the client and database, providing a secure channel to read and write the Personalization data.

Personalization Virtualization Component (PVC)
The component responsible for redirecting reads and writes of profile data from within a managed application.

Privilege Discovery
The Application Manager functionality to monitor endpoints in order to identify applications that use administrative privileges.

Process Rules
Application Manager rules that manage access for a parent process to run child processes that might be managed differently in other rules.

PVC
See Personalization Virtualization Component.
**registry hive**

A section of the registry that is a logical grouping of registry keys, subkeys and values. Registry hives are denoted by the prefix HKEY.

**regular expression**

An expression that describes or matches a set of strings. Regular expressions are usually used to give a concise description of a set without having to list all elements and to search and manipulate bodies of text based on certain patterns.

**Restricted**

The Security Level assigned to users, groups or devices in an Application Manager Privilege Discovery at which only authorized applications can run. These include files owned by members of the Trusted Owners list and files listed in Allowed Items, Trusted Vendors and Trusted Ownership.

**reusable condition**

A condition that can be used multiple times within a configuration, ideal for grouping common sets of conditions together that need to run regularly in a variety of circumstances.

**reusable node**

A node that can be used multiple times within a configuration, ideal for grouping common sets of actions together that need to run regularly in a variety of circumstances.

**Secure Hash Identifier (SHA-1)**

A cryptographic hash function designed by the United States National Security Agency and a U.S. Federal Information Processing Standard published by the United States NIST. SHA-1 produces a 160-bit (20-byte) hash value known as a message digest. A SHA-1 hash value is typically rendered as a hexadecimal number, 40 digits long.

**Security Identifier SID**

A data structure of variable length that identifies user, group, and computer accounts. Every account on a network is issued a unique SID when the account is first created. Internal processes in Windows refer to an account's SID rather than the account's user or group name. Likewise, Application Manager also refers to a user or group SID unless the SID could not be found when added to the configuration.

**Security Level**

Application Manager configuration settings that specify how to manage requests to run unauthorized applications by the users, groups, or devices that a rule matches. Security levels include Restricted, Self-Authorizing, Audit only, and Unrestricted.

**Self Heal**

Environment Manager mechanism to automatically restore environment items, including files, processes, services, or registry keys.
Self-Authorizing

The Security Level at which the user, group, or device is granted control to choose whether to block or run an unauthorized application on the host computer. The Self-Authorizing Security Level can be assigned in an Application Manager Privilege Discovery to match a file execute request for users, groups, or devices.

Server Configuration Portal (SCP)

A utility for configuring AppSense servers and databases. During product installation it is used to create server instances, databases and user accounts but can be used at any time for maintenance and troubleshooting.

SHA-1

See Secure Hash Identifier.

SHA-256

A hash function computed with 32-bit words. See also: SHA-1 and Adler-32.

SID

See Security Identifier.

Site

In Environment Manager, a logical grouping of clients and Personalization Servers communicating with a database.

TCP


Time Limits

Settings applied to entries in the Allowed Items and Denied Items nodes of an Application Manager Privilege Discovery that determine day and time ranges when the controls apply.

Transmission Control Protocol (TCP)

A standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other.

trigger

Preset User and Computer events that trigger actions and conditions.

Trusted Applications

Files that are authorized to run by an Application Manager configuration and can execute files that are normally prohibited. Trusted Applications are designated in the Default Rules and include specified Trusted Content, which includes files normally prohibited but allowed
when run executed as a child process of the associated Trusted Application. Application matching takes place when a file is prohibited by a rule or fails Trusted Ownership checking.

**Trusted Ownership**

A secure method Application Manager uses to prevent users running unauthorized applications. On NTFS formatted drives, files have owners and Application Manager is configured by default to only allow files to be executed if the file owner is a member of the Trusted Owners list. If a user tries to run a file that is not owned by a trusted owner, the execute request is denied and a message notifies the user.

**Trusted Vendors**

Digital certificates signed by trusted sources. In Application Manager, Trusted Vendor checking allows applications that fail Trusted Ownership checking to match digital certificates with the Trusted Vendors list. A list of Trusted Vendors can be defined for each User, Group, Device, Custom, Scripted, and Process rule of the configuration. DesktopNow queries each file execution that fails Trusted Ownership checking to detect the presence of a digital certificate. If the file has a digital certificate that is signed by a certificate authority matching a valid entry in the Trusted Vendor list, the file is allowed to run.

**UNC**

See Universal Naming Convention.

**Universal Naming Convention (UNC)**

A NetBIOS naming format for identifying the location of servers, printers, and other resources on a local area network (LAN). UNC begins with two backslashes (\) and takes the form \Computer_name\Share_name

**Unrestricted**

The Security Level assigned to users, groups, devices in an Application Manager Privilege Discovery at which all actions are permitted without event logging or auditing.

**User Personalization**

In Environment Manager, the ability to capture application and desktop changes on a central database and reapply saved settings for users when triggered, such as at logon or application start.

**User Privileges Management**

An Application Manager feature that provides a granular approach to delegating administrative rights to users and applications by assigning rights according to merit. This level of control can be deployed to elevate or restrict privileges on a case by case basis according to the preferred approach taken in the environment.

**User Virtualization (UV)**

The independent management of all aspects of the user on the desktop. The user’s personality is decoupled from the operating system and applications managed
independently and applied to a desktop as needed without using scripting, group policies, or user profiles, regardless of how the desktop is delivered.

**UV**

See User Virtualization.

**UV Service**

The NT Service that is responsible for managing user virtualization on the endpoint. Stopping this service will prevent the endpoint from being managed by Environment Manager.

**VDI**

See Virtual Desktop Infrastructure.

**Virtual Desktop Infrastructure (VDI)**

The server computing model enabling desktop virtualization, encompassing the hardware and software systems required to support the virtualized environment.

**Windows Management Instrumentation (WMI)**

A set of extensions to the Windows Driver Model that provides an operating system interface through which instrumented components can provide information and notification.

**Windows Script Host (WSH)**

The file format for Windows Script Host, an automation technology for Windows that provides language-independent scripting abilities similar to batch files, and is used to create logon scripts, automation, and batch files.

**WMI**

See Windows Management Instrumentation.

**WSH**

See Windows Script Host.