



Letzte Aktualisierung : 19.01.2003

Einführung in Microsoft Clusterdienst (MSCS) in Windows Server 2003

Autor : Mohan Rao Cavale

Inhalt

[Einführung](#)

[Drei Technologien für die Clusterbildung](#)

[Failoverfähigkeit über Microsoft Clusterdienst](#)

[Architektur des Clusterdiensts](#)

[Anwendung ohne Clusterunterstützung](#)

[Notepad mit hoher Verfügbarkeit](#)

[Schlussfolgerung](#)

Einführung

Die Bereitstellung einer qualitativ hochwertigen Anwendung mit zahlreichen Features ist nicht immer ausreichend - zunehmend müssen diese Anwendungen eine hohe Verfügbarkeit besitzen. Haben Sie es bisher vermieden, Ihre Anwendung weiterzuentwickeln, weil die Clustertechnologie viel zu abschreckend und kompliziert wirkte? Mit Microsoft Clusterdienst - eingeführt in Windows NT™ 4 und verfügbar in Windows Server 2003, Enterprise Edition & Datacenter Edition - können Entwickler mit unkomplizierten Tools Anwendungen in einer Clusterumgebung bereitstellen. Dazu gehört die Möglichkeit, eine beliebige Anwendung als allgemeine Anwendung in einem Cluster zu registrieren und die Anwendungskonfiguration mithilfe von Windows Scripting zu steuern.

Ein Cluster verbindet zwei oder mehrere Server, so dass sie den Clients wie ein einzelner Computer erscheinen. Die Verbindung von Servern in einem Cluster ermöglicht eine Arbeitslastverteilung, zentrale Vorgänge/Verwaltung und bietet zudem eine Skalierungsoption für steigende Anforderungen. Auf diese Weise können Sie durch die Clusterbildung Anwendungen mit hoher Verfügbarkeit entwickeln.

Dieser Artikel beschäftigt sich mit dem Clusterdienst, eine der drei Servertechnologien von Microsoft, die die Clusterbildung unterstützen. Wir zeigen, wie auf einfache Weise eine Funktionsüberprüfung für Ihre Anwendung innerhalb einer Clusterumgebung ausgeführt werden kann, ohne den Code der Anwendung ändern zu müssen.

Drei Technologien für die Clusterbildung

Auf Microsoft-Servern stehen drei Technologien zur Unterstützung der Clusterbildung bereit: Netzwerklastenausgleich (Network Load Balancing, NLB), Komponentenlastenausgleich (Component Load Balancing, CLB) und Microsoft Clusterdienst (Microsoft Cluster Service, MSCS).

Netzwerklastenausgleich

Der Netzwerklastenausgleich agiert als Front-End-Cluster, der den eingehenden IP-Datenverkehr über

einen Servercluster verteilt. Diese Technologie bietet E-Commerce-Websites eine optimale inkrementelle Skalierbarkeit und hervorragende Verfügbarkeit. Bis zu 32 Windows Enterprise Server können zur gemeinsamen Nutzung einer einzelnen virtuellen IP-Adresse verbunden werden. NLB verbessert die Skalierbarkeit durch Verteilung der Clientanforderungen über mehrere Server innerhalb des Clusters. Wenn das Datenaufkommen steigt, können dem Cluster zusätzliche Server hinzugefügt werden; bis zu 32 Server können in einem beliebigen Cluster enthalten sein. NLB bietet ebenfalls eine hohe Verfügbarkeit durch die automatische Erkennung eines Serverausfalls und Neupartitionierung des Clientdatenverkehrs zwischen den übrigen Servern innerhalb von 10 Sekunden, ohne dass der Betrieb für die Benutzer unterbrochen wird.

Komponentenlastenausgleich

Der Komponentenlastenausgleich verteilt die Arbeitslast auf mehrere Server, auf denen die Geschäftslogik einer Site ausgeführt wird. CLB erzielt einen dynamischen Ausgleich von COM+-Komponenten über eine Gruppe von bis zu acht identischen Servern. Beim CLB befinden sich die COM+-Komponenten auf den Servern in einem separaten COM+ -Cluster. Aufrufe zur Aktivierung von COM+ -Komponenten werden gleichmäßig an die verschiedenen Server innerhalb des COM+ -Clusters verteilt. CLB ergänzt den NLB und Clusterdienst, da der CLB auf der mittleren Ebene in einem Clusternetzwerk mit mehreren Ebenen agiert. CLB wird als Feature von Application Center 2000 bereitgestellt. CLB und Microsoft Clusterdienst können auf derselben Computergruppe ausgeführt werden.

Clusterdienst

Der Clusterdienst agiert als Back-End-Cluster; er bietet eine hohe Verfügbarkeit für Anwendungen wie Datenbank-, Messaging- sowie Datei- und Druckdienste. MSCS versucht, die Auswirkungen eines Ausfalls im System zu minimieren, wenn ein beliebiger Knoten (ein Server im Cluster) ausfällt oder offline geschaltet wird.

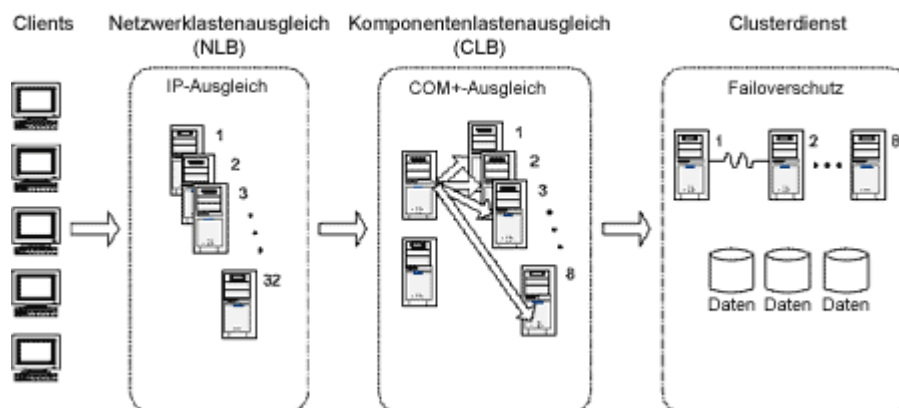


Abbildung 1. Drei Servertechnologien von Microsoft, die Clusterbildung unterstützen

Failoverfähigkeit über Microsoft Clusterdienst

Die MSCS-Failoverfähigkeit wird durch Redundanz über mehrere in einem Cluster verbundene Computer erreicht, wobei jeder Computer einen unabhängigen Fehlerstatus aufweist. Für eine Redundanz ist es erforderlich, dass die Anwendungen auf mehreren Servern innerhalb des Clusters installiert sind. Jedoch ist eine Anwendung zu einem bestimmten Zeitpunkt nur in einem Knoten aktiv. Wenn ein Problem mit dieser Anwendung auftritt oder der Server heruntergefahren wird, wird die Anwendung in einem anderen

Knoten neu gestartet. Windows Server 2003 Datacenter Edition unterstützt bis zu 8 Knoten in einem Cluster.

Jeder Knoten verfügt jeweils über Speicher, Systemfestplatte, Betriebssystem und einen Teilsatz der Clusterressourcen. Wenn ein Knoten ausfällt, übernimmt ein anderer Knoten die Besitzrechte an den Ressourcen des ausgefallenen Knotens (dieser Vorgang wird als "Failover" bezeichnet). Microsoft Clusterdienst registriert dann die Netzwerkadresse für die Ressource auf dem neuen Knoten, so dass der Clientdatenverkehr an das verfügbare System weitergeleitet wird und die Besitzrechte an der Ressource erhalten hat. Wenn die ausgefallene Ressource zu einem späteren Zeitpunkt wieder online geschaltet wird, kann MSCS so konfiguriert werden, dass die Ressourcen und Clientanforderungen entsprechend neu verteilt werden (dieser Vorgang wird als "Failback" bezeichnet). Damit die Anwendung an dem Zeitpunkt fortgesetzt werden kann, an dem der Failover aufgetreten ist, müssen die Knoten Zugriff auf einen freigegebenen Speicher haben, in dem der Anwendungsstatus gespeichert ist.

Beachten Sie, dass Microsoft Clusterdienst zur Bereitstellung einer hohen Verfügbarkeit ausgelegt wurde und nicht für eine Gewährleistung einer echten Fehlertoleranz. Der Begriff "fehlertolerant" wird im Allgemeinen verwendet, um eine Technologie zu beschreiben, die in höherem Maße eine Ausfallsicherheit und Wiederherstellung bietet. Fehlertolerante Server verwenden in der Regel hochwertige Hardware oder Datenredundanz zusammen mit spezialisierter Software, um eine fast sofortige Wiederherstellung bei Hardware- oder Softwareausfällen zu gewährleisten. Diese Lösungen sind weitaus kostspieliger als eine Clusterlösung, da Sie für redundante Hardware zahlen müssen, die nur dann eingesetzt wird, wenn bei einem Ausfall eine Wiederherstellung ausgeführt werden muss. Microsoft Clusterdienst bietet eine sehr gute Lösung für eine hohe Verfügbarkeit, indem kostengünstige Standardhardware eingesetzt und gleichzeitig Computerressourcen maximal genutzt werden.

Microsoft Clusterdienst basiert auf dem Clustermodell "Freigegeben - Nichts". Das Modell "Freigegeben - Nichts" schreibt vor, dass obgleich mehrere Knoten im Cluster Zugriff auf ein Gerät oder eine Ressource haben, diese Ressource nur einem System gehört und nur von einem System zu einem bestimmten Zeitpunkt verwaltet wird. (In einem MSCS-Cluster wird eine Ressource als eine physische oder logische Komponente definiert, die online oder offline geschaltet, in einem Cluster verwaltet, nur von einem Knoten gehostet und zwischen den Knoten verschoben werden kann.)

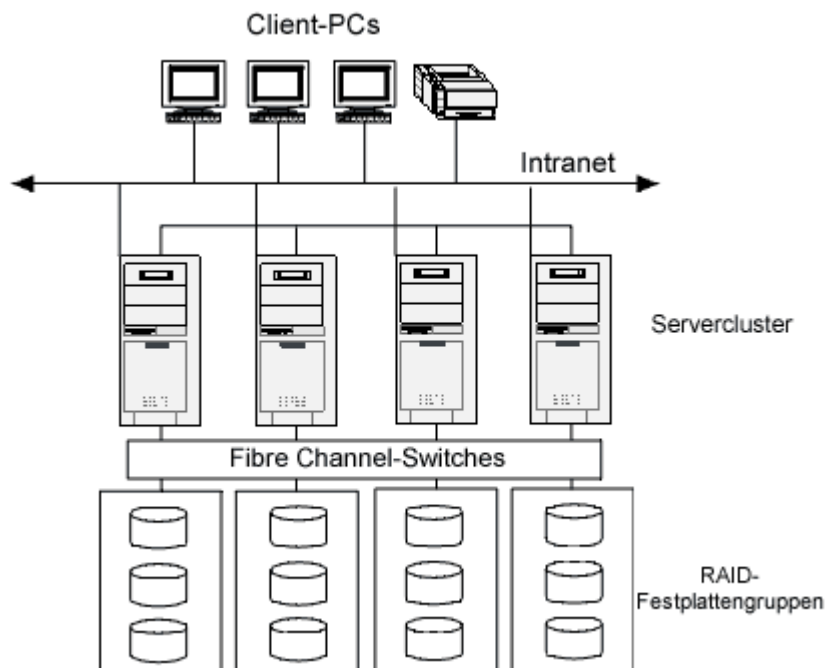


Abbildung 2. Microsoft Clusterdienst

Architektur des Clusterdiensts

Microsoft Clusterdienst besteht aus drei Hauptkomponenten: Clusterdienst, Ressourcenmonitor und Ressourcen-DLLs. Zusätzlich können mit Cluster Administrator (Clusterverwaltung) Erweiterungs-DLLs für Verwaltungsaufgaben erstellt werden.

Clusterdienst

Der Clusterdienst ist die Kernkomponente und wird als Systemdienst mit hoher Priorität ausgeführt. Der Clusterdienst steuert Clusteraktivitäten und führt Aufgaben wie die Koordination der Ereignisbenachrichtigung, Erleichterung der Kommunikation zwischen Clusterkomponenten, Abwicklung von Failovervorgängen und Konfigurationsverwaltung durch. Jeder Clusterknoten führt seinen eigenen Clusterdienst aus.

Ressourcenmonitor

Der Ressourcenmonitor ist eine Schnittstelle zwischen dem Clusterdienst und den Clusterressourcen und wird als eigenständiger Prozess ausgeführt. Der Clusterdienst verwendet den Ressourcenmonitor zur Kommunikation mit den Ressourcen-DLLs. Die DLL ist für die gesamte Kommunikation der Ressourcen verantwortlich. Wenn also die DLL in einem Ressourcenmonitor gehostet wird, wird der Clusterdienst von allen Ressourcen abgeschirmt, die nicht ordnungsgemäß bzw. gar nicht funktionieren. In einem einzelnen Knoten können mehrere Instanzen des Ressourcenmonitors ausgeführt werden, wodurch unberechenbare Ressourcen von anderen Ressourcen abgeschirmt werden können.

Wenn der Clusterdienst einen Vorgang auf einer Ressource ausführen muss, sendet er eine Anforderung an den Ressourcenmonitor, der dieser Ressource zugewiesen ist. Sollte der Ressourcenmonitor im Prozess keine DLL enthalten, die diesen Ressourcentyp verarbeiten kann, verwendet er die Registrierungsinformationen, um die mit diesem Ressourcentyp verbundene DLL zu laden. Anschließend übergibt er die Anforderung des Clusterdiensts an eine der Einstiegspunktfunktionen der DLL. Die Ressourcen-DLL ist für die Vorgangsdetails verantwortlich, das heißt für die besonderen Anforderungen der Ressource.

Ressourcen-DLL

Die dritte Hauptkomponente von Microsoft Clusterdienst ist die Ressourcen-DLL. Der Ressourcenmonitor und die Ressourcen-DLL kommunizieren mithilfe der Ressourcen-API miteinander. Diese Ressourcen-API ist eine Sammlung von Einstiegspunkten, Rückruffunktionen und dazugehörigen Strukturen sowie Makros zur Ressourcenverwaltung.

Für den Clusterdienst ist eine Ressource eine beliebige physische oder logische Komponente, die verwaltet werden kann. Beispiele für Ressourcen sind Datenträger, Netzwerknamen, IP-Adressen, Datenbanken, Websites, Anwendungen sowie jede andere Einheit, die online oder offline geschaltet werden kann. Ressourcen werden nach Typ aufgeteilt. Zu den Ressourcentypen gehören physische Hardware (z.B. Laufwerke) und logische Elemente (z.B. IP-Adressen, Dateifreigaben und allgemeine Anwendungen).

Jede Ressource verwendet eine Ressourcen-DLL, eine weitgehend passive Konvertierungsebene zwischen Ressourcenmonitor und Ressource. Der Ressourcenmonitor ruft die Einstiegspunktfunktionen der Ressourcen-DLL auf, um den Status der Ressource zu überprüfen und die Ressource online und offline zu schalten. Die Ressourcen-DLL ist für die Kommunikation mit ihrer Ressource über jede geeignete IPC-Methode zur Implementierung dieser Methoden verantwortlich.

Anwendungen, die ihre eigenen Ressourcen-DLLs zur Kommunikation mit dem Clusterdienst implementieren und die Cluster-API verwenden, um Clusterinformationen anzufordern und zu aktualisieren, werden als Anwendungen mit Clusterunterstützung bezeichnet. Anwendungen und Dienste, die keine Cluster- oder Ressourcen-APIs und Clustersteuerungsfunktionen verwenden, unterstützen keine Clusterbildung und erkennen den Clusterdienst nicht. Diese Anwendungen ohne Clusterunterstützung werden in der Regel als allgemeine Anwendungen oder -dienste verwaltet.

Anwendungen mit und ohne Clusterunterstützung werden auf einem Clusterknoten ausgeführt und können als Clusterressourcen verwaltet werden. Jedoch können nur Anwendungen mit Clusterunterstützung die vom Clusterdienst bereitgestellten Features über die Cluster-API nutzen. Für die Entwicklung einer Anwendung mit Clusterunterstützung ist die Erstellung eines benutzerdefinierten Ressourcentyps erforderlich. Mit dem benutzerdefinierten Ressourcentyp können die Entwickler ihre Anwendungen so gestalten, dass sie entsprechend auf unterschiedliche im Cluster auftretende Ereignisse antworten und reagieren können (z.B. der Knoten wird offline geschaltet, folglich wird die Datenbankverbindung getrennt).

Für die meisten Anwendungen, die in einem Cluster ausgeführt werden müssen, ist es wünschenswert, dass Zeit und Ressourcen aufgewendet werden, um einen benutzerdefinierten Ressourcentyp zu entwickeln. Jedoch können Anwendungen zunächst innerhalb einer Clusterumgebung getestet werden, ohne dass der Anwendungscode geändert oder ein neuer Ressourcentyp erstellt werden muss. Innerhalb von Windows Server 2003 können unveränderte Anwendungen auf Basisebene Teil von Anwendungen "ohne Clusterunterstützung" sein. Der Clusterdienst stellt den Ressourcentyp "Generic Application" (Allgemeine Anwendung) für diesen Zweck bereit.

Erweiterungs-DLL für Cluster Administrator

Die Erweiterungs-DLL für Cluster Administrator (Clusterverwaltung) bietet anwendungsspezifische Verwaltungsfeatures innerhalb von Cluster Administrator, mit denen die Benutzer ihre Anwendungen unabhängig davon, ob die Anwendung innerhalb oder außerhalb eines Clusters ausgeführt wird, auf die gleiche Weise verwalten können. Innerhalb der Cluster Administrator-Umgebung können die Entwickler Anwendungsverwaltungsfeatures oder einfach einen Link zu einem vorhandenen Verwaltungstool bereitstellen.

Die Entwickler erweitern die Funktionalität von Cluster Administrator, indem sie eine Erweiterungs-DLL schreiben. Die Cluster Administrator-Anwendung kommuniziert mit den Erweiterungs-DLLs über einen festgelegten Satz von COM-Schnittstellen. Die Erweiterungs-DLLs müssen einen bestimmten Schnittstellensatz implementieren und auf jedem Clusterknoten registriert sein.

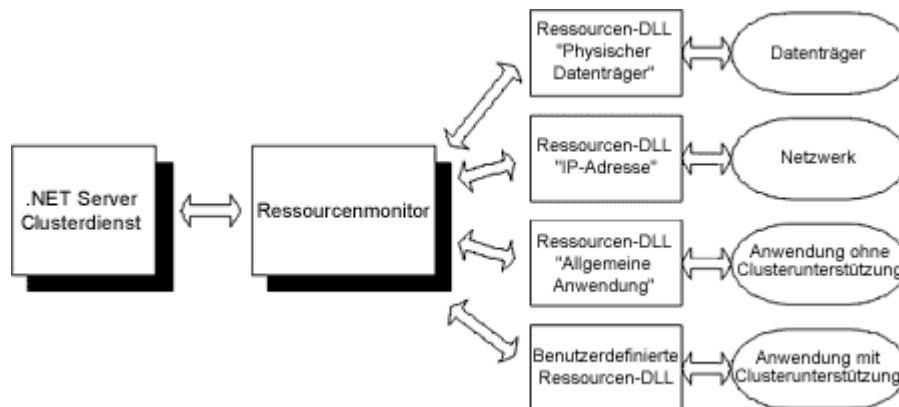


Abbildung 3. Hauptkomponenten: Clusterdienst, Ressourcenmonitor und Ressourcen-DLLs

Anwendung ohne Clusterunterstützung

Anwendungen oder Dienste, die nicht ihre eigenen Ressourcen -DLLs bereitstellen, können dennoch in der Clusterumgebung konfiguriert werden. Der Clusterdienst in Windows Server 2003 umfasst eine allgemeine Ressourcen-DLL für diesen Zweck - die Ressourcen -DLL für allgemeine Anwendungen und die Ressourcen-DLL für allgemeine Dienste. Der Clusterdienst behandelt diese Anwendungen bzw. Dienste als allgemeine Anwendungen oder Dienste ohne Clusterunterstützung.

Die allgemeine Ressourcen-DLL bietet nur sehr grundlegende Steuerungsmöglichkeiten. Die Ressourcen-DLL für allgemeine Anwendungen sucht z.B. nach Anwendungsausfällen, indem sie feststellt, ob der Anwendungsprozess noch vorhanden ist. Anschließend wird die Anwendung durch Beenden des Prozesses offline geschaltet. Zu den anderen Ressourcen bestehen jedoch keine Abhängigkeiten, und sie bietet eine unkomplizierte Methode zum Testen einer Anwendung in einer Clusterumgebung.

Notepad mit hoher Verfügbarkeit

Nicht alle Anwendungen werden in einem Cluster effektiv funktionieren. Die wirksamste Auswertungsmethode ist die tatsächliche Bereitstellung der Anwendung in einem Cluster. Für einen ersten Testlauf sollten Sie Ihre Anwendung mithilfe des integrierten Ressourcentyps "Generic Application" (Allgemeine Anwendung) in einem Cluster registrieren. Der Ressourcentyp "Generic Application" wird als Teil des Clusterdiensts in Windows Server 2003 bereitgestellt und kann in Cluster Administrator (Clusterverwaltung) unter Cluster Configuration (Clusterkonfiguration) über den **Resource Types**-Knoten (Ressourcentypen) zusammen mit den anderen integrierten Ressourcentypen angezeigt werden (siehe Abbildung 4).

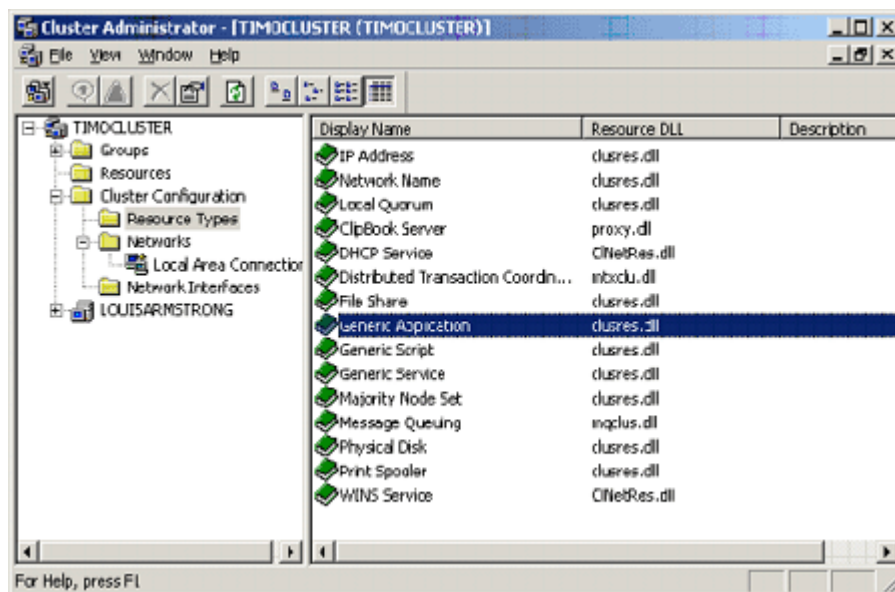


Abbildung 4. "Resource Types"-Knoten in Cluster Administrator

Cluster Administrator verfügt über interaktive Assistenten, mit denen Sie Ressourcen für alle aufgeführten Ressourcentypen erstellen können. Der Clusterdienst bietet jedoch auch COM-Schnittstellen zur programmtechnischen Erstellung und Verwaltung von Ressourcen.

Anmerkung:

Das neueste Cluster Administrator-Tool und dazugehörige Entwicklungsressourcen können über das Plattform-SDK bezogen werden.

Cluster Automation Server

Cluster Automation Server bietet eine Reihe von Automatisierungsobjekten, die eine vollständige Clusterverwaltungsschnittstelle für Skriptsprachen darstellen, wodurch Sie webbasierte Remoteverwaltungstools entwickeln können. Cluster Automation Server vereinfacht und verbessert den Prozess für die Erstellung einer Clusterverwaltungsanwendung.

Die objektorientierte Struktur von Cluster Automation Server beinhaltet, dass bei fast allen Clusterprogrammieraufgaben die folgenden Schritte berücksichtigt werden müssen:

1. Festlegen des auszuführenden Clustervorgangs.
2. Suchen nach dem **Cluster Automation Server**-Objekt, das über die Eigenschaft bzw. Methode zur Ausführung des Vorgangs verfügt.
3. Feststellen, wie das Objekt aus Schritt 2 abgerufen werden kann. Die in MSDN® verfügbare Objekthierarchie ist dafür nützlich.
4. Abrufen des Objekts und Aufrufen der Eigenschaft bzw. Methode.

Zur Veranschaulichung verwenden wir den Windows Scripting Host und Microsoft VBScript, um unsere Ressource "Generic Application" programmtechnisch zu erstellen.

Clusterobjekt

Das Clusterobjekt ist ein Objekt auf oberster Ebene, das die Erstellung neuer Instanzen ermöglicht. Die ProgID für das Clusterobjekt lautet "MSCLUSTER.CLUSTER":

```
Set oCluster = CreateObject( "MSCluster.Cluster" )
```

Öffnen des Clusters

Bevor Sie eine Methode in einem Cluster verwenden, muss eine Verbindung zu diesem Cluster geöffnet werden. Die **Open**-Methode öffnet eine Verbindung zu einem Cluster. Die Übergabe einer leeren Zeichenfolge ("") als Parameter an die **Open**-Methode öffnet eine Verbindung zum Cluster auf dem lokalen Host. Unser Skript wird auf dem lokalen Hostserver ausgeführt:

```
oCluster.Open( "" )
```

Erstellen einer Gruppe

Eine Clustergruppe ist ein Container für Clusterressourcen. Wenn eine Ressource in einer Gruppe ausfällt und die Ressource in einen anderen Knoten verschoben werden muss, werden alle Ressourcen in der Gruppe verschoben. Durch Gruppen sind auch die Abhängigkeitsgrenzen festgelegt. Eine Ressource kann nur eine Abhängigkeit mit einer anderen Ressource aufweisen, wenn sich diese Ressource in derselben Gruppe befindet. Für unseren Test erstellen wir eine eindeutige Gruppe mit dem Namen "Notepad mit hoher Verfügbarkeit":

```
Set oGroup = oCluster.ResourceGroups.CreateItem( "Notepad mit hoher Verfügbarkeit" )
```

Erstellen der Ressource

Jede Gruppe enthält eine Sammlung von Ressourcen. Die Methode **CreateItem** erstellt eine neue Ressource und fügt sie der Gruppensammlung hinzu. In unserem Beispiel erstellen wir eine Ressource mit

dem Namen "Notepad" vom Ressourcentyp "Generic Application":

```
Set oResource = oGroupResources.CreateItem( "Notepad", "Generic Application ", 0 )
```

Festlegen der Ressourceneigenschaften

Jede Ressource vom Typ "Generic Application" verfügt über zwei Eigenschaften, die wichtig sind, um die Ressource online zu schalten: **CommandLine** und **CurrentDirectory**. **CommandLine** enthält den Befehl, der ausgeführt wird, wenn die Ressource online geschaltet wird; **CurrentDirectory** gibt das Systemverzeichnis an, in dem der Befehl ausgeführt wird. Wenn dieses Skript die Anweisung ausführt, die Ressource online zu schalten, wird Notepad gestartet. Damit Notepad angezeigt wird, haben wir die Eigenschaft **InteractWithDesktop** auf **1** gesetzt.

```
Set oProperties = oResource.PrivateProperties

' Eigenschaften der allgemeinen Anwendung festlegen
oProperties.Item("CommandLine") = "notepad"
oProperties.Item("CurrentDirectory") = "javascript:void(null);"
oProperties.Item("InteractWithDesktop") = 1

oProperties.SaveChanges
```

Onlineschalten der Ressource

Mit der **Online**-Methode wird eine Ressource online geschaltet. Online ist ein Status, der besagt, dass die Ressource für den Cluster verfügbar ist. Für unsere allgemeine Anwendung bedeutet das Onlineschalten der Ressource, dass Notepad gestartet wird.

```
oResource.Online 10
```

Vollständiges Skript

```
Option Explicit

Main
.....
Hauptsubroutine
.....
Sub Main

    Dim oGroup
    Dim oCluster
    Dim oResource

    ' Clusterobjekt erstellen.
    Set oCluster = CreateObject( "MSCluster.Cluster" )

    ' Cluster öffnen. Eine leere Zeichenfolge bedeutet, dass der lokale Cluster geöffnet wird.
    oCluster.Open( "" )

    ' Gruppe erstellen oder öffnen.
    AddGroup oCluster, oGroup

    ' Ressource erstellen oder öffnen.
    AddResource oGroup, oResource

    ' Ressource online schalten und bis zu 10 Sekunden warten,
    ' bis sie online ist.
    oResource.Online 10

End Sub

.....
' Diese Subroutine erstellt oder öffnet die Gruppe.
.....
Sub AddGroup( oCluster, oGroup )
    Set oGroup = oCluster.ResourceGroups.CreateItem( "Notepad mit hoher
        Verfügbarkeit" )
End Sub
```



```

.....
Diese Subroutine fügt der Gruppe die Ressource hinzu.
.....
Sub AddResource( oGroup, oResource )
    Dim oGroupResources
    Dim oProperties
    Dim oCLProperty
    Dim oCDProperty

    Set oGroupResources = oGroup.Resources
    Set oResource = oGroupResources.CreateItem( "Notepad", "Generic Application
", 0 ) 'CLUSTER_RESOURCE_DEFAULT_MONITOR
    Set oProperties = oResource.PrivateProperties

    ' Eigenschaften der allgemeinen Anwendung festlegen
    oProperties.Item("CommandLine") = "notepad"
    oProperties.Item("CurrentDirectory") = "javascript:void(null);"
    oProperties.Item("InteractWithDesktop") = 1

    oProperties.SaveChanges
End Sub

.....
' Festgelegten Zeitraum warten.
.....
Sub Sleep( PauseTime )
    Dim Start

    Start = Timer

    Do While Timer < Start + PauseTime
    Loop

End Sub

```

Durch Ausführen dieses einfachen Skripts wird unsere neue Ressource "Notepad" erstellt und in ihrer eigenen Gruppe platziert ("Notepad mit hoher Verfügbarkeit").

Überprüfen der Ergebnisse

Mithilfe von Cluster Administrator können wir unsere Ergebnisse überprüfen. Wir sehen, dass die Parameter korrekt gesetzt wurden, indem wir uns die Eigenschaften der Notepad-Ressource in Cluster Administrator anzeigen (siehe Abbildung 5).

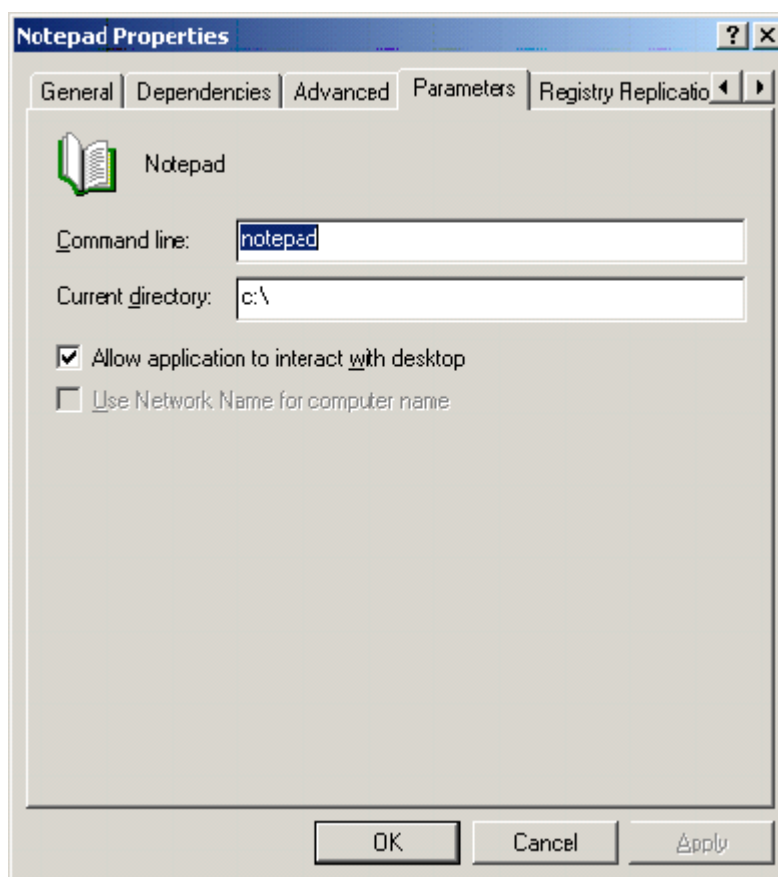


Abbildung 5. Ressourcenparameter in Notepad

Auf der Registerkarte **Advanced** sehen Sie die Standardeigenschaften, die besagen, dass die Anwendung bei einem Fehler bis zu 3 Mal neu gestartet werden soll. Die Abrufintervalle **LooksAlive** und **IsAlive** werden standardmäßig auf die Werte des Ressourcentyps gesetzt, können jedoch durch Angabe eines anderen Werts außer Kraft gesetzt werden. Da diese Anwendung keinen gesonderten Code enthält, durch den sie eine Clusterunterstützung erhält, wird die Anwendung nur dann als aktiv angesehen, wenn im System dieser Anwendungsprozess ausgeführt wird.

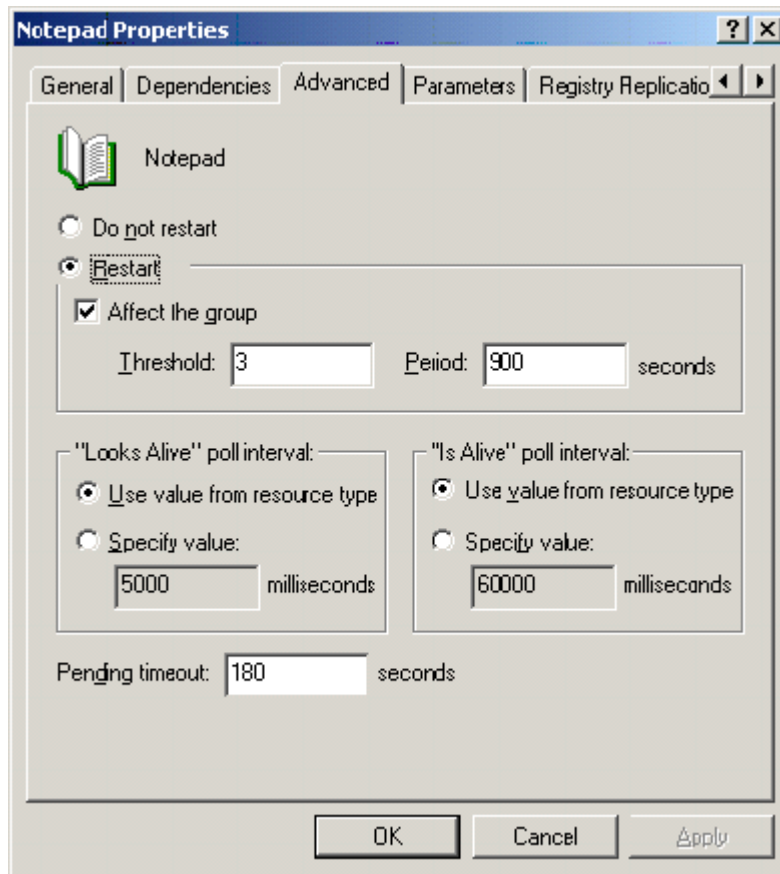


Abbildung 6. Registerkarte "Advanced" der Notepad-Ressource

Testen der Anwendung

Wenn Sie die Notepad-Ressource online schalten, wird auf dem Server Notepad gestartet. Sollte der Notepad-Prozess beendet werden, wird er sofort wieder gestartet. Dies zeigt den Clusterdienst in Aktion - Ihre Anwendung bleibt aktiv und wird ohne Unterbrechung ausgeführt. Als Ressource "Generic Application" erkennt der Clusterdienst, ob ein Anwendungsprozess nicht mehr ausgeführt wird und kann dann entsprechende Korrekturmaßnahmen durchführen.

Was geschieht jedoch, wenn unsere Anwendung auf eine Weise ausfällt, die nicht zu einer Beendigung des Prozesses führt (beispielsweise durch einen Netzwerkausfall, Stillstand oder eine Threadbeendigung im Hintergrund)? Leider bietet der Ressourcentyp "Generic Application" nur eine generische Fehlererkennung. Die meisten Entwickler, die Anwendungen schreiben, die in einer Clusterumgebung ausgeführt werden, bevorzugen benutzerdefinierte Ressourcen -DLLs, um mit anwendungsspezifischen Problemen umzugehen. Der Ressourcentyp "Generic Application" ist jedoch zur schnellen Auswertung Ihrer Anwendung in einem Cluster unschlagbar.

Schlussfolgerung

Microsoft Clusterdienst bietet eine hohe Verfügbarkeit, indem kostengünstige Standardhardware eingesetzt und gleichzeitig Computerressourcen maximal genutzt werden. Der Clusterdienst in Windows Server 2003 stellt leistungsfähige Tools bereit, damit Ihre Anwendungen eine hohe Verfügbarkeit erhalten. Das Schreiben von Anwendungen mit Clusterunterstützung mag für einige Entwickler als zu kostenintensiv oder abschreckend erscheinen. Um den Entwicklern die Vorteile der Clusterbildung mit nur wenig Aufwand vorab aufzuzeigen, stellt der Clusterdienst den Ressourcentyp "Generic Application"

bereit, mit dem Anwendungen problemlos für eine Ausführung innerhalb eines Clusters konfiguriert werden können. Obwohl mit dem Ressourcentyp "Generic Application" vielleicht nicht der Vollkommenheitsgrad erreicht wird, der für eine Produktionsanwendung erforderlich ist, bietet er eine Testmöglichkeit für das Verhalten Ihrer Anwendung innerhalb eines Clusters.

Alle mit ® und ™ gekennzeichneten Bezeichnungen sind Marken oder eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Alle anderen Bezeichnungen sind Marken oder eingetragene Marken der jeweiligen Hersteller.

<http://www.microsoft.com/germany/misc/copyright.htm>