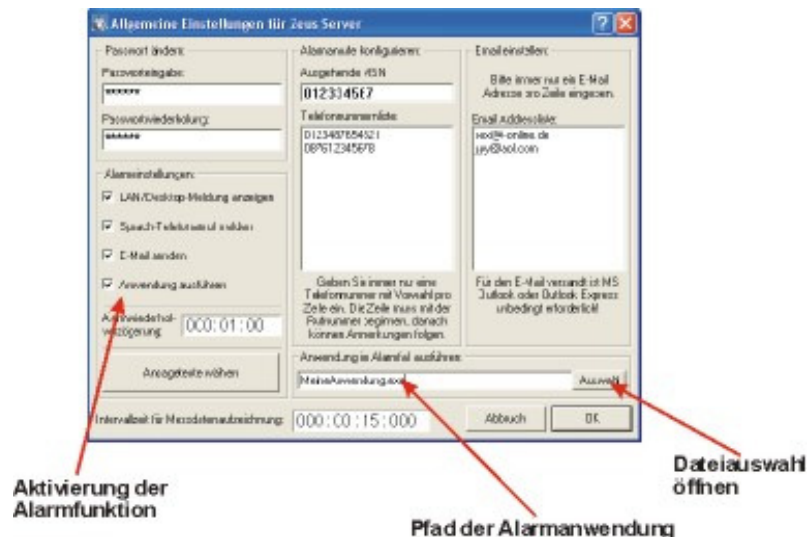


Kurzbeschreibung für die Ausführung von Alarmanwendungen mit dem Programm Zeus Serverraumüberwachung ab Version 2.9.00:

Ab der Version V2.9.00 verfügt die Zeus Applikation für Serverraumüberwachung über eine Schnittstelle, die es erlaubt fast beliebige Anwendungsprogramme oder selbst erstellte DLL's im Falle eines Alarmzustandes aufzurufen.

Zur Aktivierung der Funktion müssen verschiedene Bedingungen erfüllt sein. Zum Einen muss die Funktion im Konfigurationsdialog aktiviert werden, zum Anderen muss die angegebene Anwendungsdatei oder DLL (mit Alarmfunktion) verfügbar sein.



Das Zeus Programm unterstützt dabei zwei unterschiedliche Schnittstellen zum Starten der Alarmanwendung, eine Applikationsschnittstelle und eine DLL Schnittstelle. Aus diesem Grund muss der Dateityp (Extension) in der Pfadangabe immer enthalten sein. Es kann entweder eine Datei des Typs „.exe“ oder eine Datei des Typs „.dll“ angegeben werden. Zeus entscheidet anhand des Dateityps, welches Interface beim Starten der Alarmanwendung verwendet wird.

1. Applikationsinterface für Dateityp „.exe“:

Bei Angabe dieses Dateityps erstellt Zeus im Alarmfall eine ASCII-Textdatei im Protokoll-Ordner, welche den Text der Alarmmeldung enthält. Dann wird die eigentliche Anwendung gestartet (unter Verwendung der Windows Funktion „ShellExecute()“). Beim Start der Anwendung wird als Kommandozeile der Dateipfad der ASCII-Datei mit der Alarmmeldung übergeben. Die Windows-Funktion berücksichtigt auch die Pfade, die in den Umgebungsvariablen angegeben sind, es sollte jedoch besser immer der komplette Pfad angegeben werden, um Missverständnisse zu vermeiden.

Beispiel für die Ausführung einer einfachen Alarmanwendung: „Notepad.exe“

2. DLL Interface für Dateityp „.DLL“

Alternativ kann eine Datei des Typs „.dll“ als Alarmanwendung bestimmt werden. Geschieht dies, dann versucht Zeus im Alarmfall diese DLL zu laden und sucht bei Erfolg die Funktion „VOID CALLBACK HandleAlert(LPSTR lpszText)“ in der DLL. Wird sie gefunden, dann ruft Zeus diese Funktion auf und übergibt als Parameter einen Zeiger auf den Text der Alarmmeldung. Diese Schnittstelle ermöglicht es dem Anwender, eine eigene Alarmverwaltung zu programmieren, welche mit jeder Programmiersprache erstellt werden kann, die in der Lage ist 32-Bit Windows DLL's zu erstellen. Zum Testen der Alarmfunktion wurde die zum Zeus Programm gehörende DLL „SensorSystems.dll“ mit einer einfachen Alarmfunktion erweitert, welche den Alarmtext in einer Message Box anzeigt.

Beispiel für die Ausführung einer einfachen Alarm-DLL: „SensorSystems.dll“

Die Testfunktion in der DLL „SensorSystems.dll“ ist folgendermaßen implementiert (Programmiersprache „C“):

```
VOID CALLBACK HandleAlert(LPSTR lpszText)
{
    if(IsBadReadPtr(lpszText, 1))          /* wenn Pointer nicht OK */
    {
        lpszText = "Es ist ein allgemeiner Alarmzustand eingetreten";
    }

    MessageBox(0L,                          /* Meldung anzeigen */
               lpszText,
               "Alarmzustand",
               MB_ICONINFORMATION | MB_OK);
}
```

Sie können eine vergleichbare Funktion in Ihrer eigenen DLL implementieren, die dann die gewünschte Alarmfunktion ausführt.

Die DLL wird bis zum Programmende von Zeus nicht mehr entladen nachdem sie einmalig aufgerufen wurde, um eventuelle Probleme mit freigegebenen Ressourcen zu vermeiden.

Für den Start der Anwendung oder DLL verwendet Zeus außerdem einen eigenen Thread, sodass die Zeus-Funktionen während der Alarmbehandlung möglichst wenig beeinflusst werden.

Bedenken Sie unbedingt, dass eine in den Adressbereich der Zeus Anwendung geladene DLL grundsätzlich in der Lage ist, bei einem funktionellen Fehler die Funktion von Zeus zu beeinflussen oder einen Programmabsturz zu verursachen. Prüfen sie Ihre selbst erstellen DLL's unbedingt sehr sorgfältig bevor Sie diese einsetzen und testen Sie die Funktion zuerst mit der Probealarm-Funktion von Zeus!