

# Repräsentation einer GUID

Bei einer GUID (Global Unique Identifier) handelt es sich um eine Zeichenfolge, die verwendet wird, um Objekte eindeutig zu bezeichnen. Bekannt geworden sind GUIDs im Rahmen der COM-Technologie (Component Object Model), um COM-Objekte eindeutig zu identifizieren.

Die Erzeugung einer GUID kann durch Funktionen des Betriebssystems, wie `CreateGuid()`, `CoCreateGuid()` oder durch Funktionen der Base Class Library (BCL) des Microsoft .NET Frameworks erfolgen. Bei der Generierung einer GUID werden Informationen des lokalen Computers verwendet, um die Eindeutigkeit zu gewährleisten. Ein Baustein hierbei ist das Identifizierungsmerkmal einer Netzwerkkarte, das als MAC-Adresse (Media Access Control Adress) bezeichnet wird. Weitere Bausteine werden aus dem aktuellen Datum und der aktuellen Uhrzeit gebildet. Zum Generieren von GUIDs existieren eine Vielzahl von Tools. Eines davon ist der Microsoft GUID Generator (`guidgen.exe`) der mit allen Versionen von Visual Studio ausgeliefert wird.

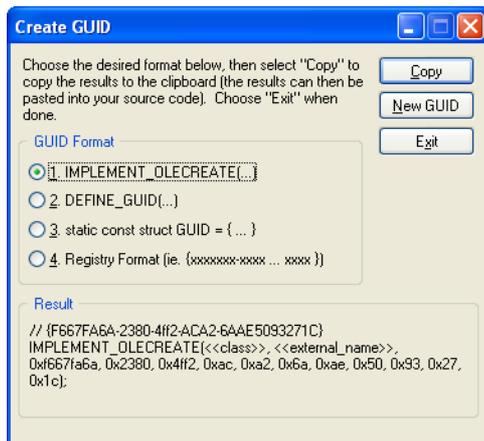


Abbildung 1: Microsoft GUID Generator

Der Microsoft GUID Generator ermöglicht die Erzeugung von GUIDs und die Präsentation in unterschiedlichen Darstellungsformen. Der Windows Installer verwendet unterschiedliche Repräsentationsformen einer GUID, die jedoch von der als »Registry Format« bezeichneten Form abgeleitet werden können.

## Standard-GUID

Bei der Standarddarstellung einer GUID handelt es sich um eine aus 38 Zeichen bestehende Zeichenfolge, bei der jedes Zeichen einen hexadezimalen Wert darstellt. Die GUID »{ 012F8BAC-80EA-43fc-BA96-CB6FFBE952A1}« wurde mit dem Microsoft GUID Generator erzeugt und entspricht augenscheinlich der Standarddarstellung einer GUID für den Windows Installer. Hierbei ist zu beachten, dass der Microsoft GUID Generator sowohl Groß- als auch Kleinbuchstaben zur Repräsentation der GUID verwendet. Der Windows Installer benötigt jedoch zwingend Großbuchstaben, so dass die dargestellte GUID entsprechend geändert werden muss »{ 012F8BAC-80EA-43FC-BA96-CB6FFBE952A1}«.

Eine Standard-GUID wird in der Systemregistrierung beispielsweise unter »HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall« verwendet, um die Produkte zu speichern, die im Dialogfeld Software der Systemsteuerung angezeigt werden.

## Gepackte GUID

Bei einer gepackten GUID handelt es sich um eine aus 32 Zeichen bestehende Darstellung einer GUID, wobei einige Zeichen entfernt und neu geordnet wurden. Nachfolgend finden Sie die bereits oben dargestellte GUID in gepackter Darstellungsform: »CAB8F210AE08CF34AB69BCF6BF9E251A«

Um eine Standard-GUID »{ 0B533DB3-A248-4E72-B47B-34F9F3342418}« in eine GUID in gepackter Darstellungsform umzuwandeln, gehen Sie wie folgt vor:

- b Entfernen Sie die öffnende und schließende Klammer: »0B533DB3-A248-4E72-B47B-34F9F3342418«
- b Verwenden Sie den ersten Zeichenblock »0B533DB3« in umgekehrter Reihenfolge: »3BD335B0«
- b Verwenden Sie den zweiten Zeichenblock »A248« in umgekehrter Reihenfolge: »842A«
- b Verwenden Sie den dritten Zeichenblock »4E72« in umgekehrter Reihenfolge: »27E4«
- b Verwenden Sie alle weiteren Zeichen »-B47B-34F9F3342418« paarweise in umgekehrter Reihenfolge, wobei Sie die Bindestriche (-) ignorieren: »4BB7439F3F434281«

- b Fassen Sie nun alle umgewandelten Zeichenblöcke zusammen, um die gepackte GUID »3BD335B0842A27E44BB7439F3F434281« zu erhalten.

Gepackte GUIDs werden in der Systemregistrierung beispielsweise unter »HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\Installer\Products« verwendet, um Informationen für Produkte abzulegen, die für alle Benutzer installiert wurden.

## Komprimierte GUID

Bei einer komprimierten GUID handelt es sich um eine aus 20 Zeichen bestehende Darstellung einer GUID, wobei die Originalzeichen ersetzt werden. Eine komprimierte GUID wird auch als »Squished GUID (zerquetschte GUID)« oder »SQUID« bezeichnet. Der Algorithmus zum Umwandeln einer SQUID in eine GUID ist nicht dokumentiert. Es existiert jedoch die undokumentierte Windows Installer-Funktion `MsiDecomposeDescriptor()` mit der Sie diese Umwandlung vornehmen können. Die Funktion wird normalerweise dazu verwendet, eine verschlüsselte Beschreibung der Windows Installer-Komponenten in seine einzelnen Bestandteile zu zerlegen. Ich werde diese Funktion an späterer Stelle in diesem Kapitel beschreiben.

Die Umwandlung der GUID »{012F8BAC-80EA-43FC-BA96-CB6FFBE952A1}« ergibt die SQUID »7HIH!\$RBq9`O-xKW14q[«. Auf der beiliegenden CD finden Sie die Anwendung »Squid2Guid.exe«, die Ihnen die Umwandlung demonstriert.

## Darwin Descriptor

Bei der Installation von Anwendungen werden Informationen auf dem Zielsystem abgelegt, die der Windows Installer benötigt um die Produktkonfiguration vorzunehmen. Verwenden Sie beispielsweise die Funktion `MsiProvideQualifiedComponent()` des Windows Installer-API können Sie den Pfad zu einer qualifizierten Windows Installer-Komponente abrufen. Die hierfür benötigten Informationen werden an unterschiedlichen Stellen in der Systemregistrierung, wie »HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\Installer\Components« abgelegt. An dieser Stelle finden Sie für jede qualifizierte Windows Installer-Komponente einen entsprechenden Unterschlüssel, der als gepackte GUID abgebildet ist. Jeder dieser Schlüssel verfügt über einen oder mehrere Werte, die die jeweilige Komponente identifizieren. Dieser Wert wird in verschlüsselter Form dargestellt und enthält Referenzen auf das zugeordnete Produkt, das zugeordnete Feature und die Komponente selbst. Die hier gewählte Darstellungsform wird als Darwin Descriptor bezeichnet.

Eine typische Darstellungsform für den Darwin Descriptor hat folgendes Erscheinungsbild: »v[X,qfU0E?q35RF2Nru?dotNET\_Framework\_SDK>HDI1hIAB\*Av(Q&g3&VT!«. Sie erkennen in der Mitte der Zeichenfolge, den Namen des Features (`dotNET_Framework_SDK`). Der zugehörige `ProductCode` und die `ComponentId` sind durch komprimierte GUIDs dargestellt. Das definitive Format des Darwin Descriptors zur Referenzierung einer Windows Installer-Komponente, ist abhängig von der Anzahl der Komponenten und Features im Installationspaket.

Anzahl der Komponenten	Anzahl der Features	Formt des Darwin Descriptors
> 1	> 1	{Komprimierter ProductCode}{Name des Features}>{Komprimierte ComponentId}
> 1	1	{Komprimierter ProductCode}>{Komprimierte ComponentId}
1	> 1	{Komprimierter ProductCode}{Name des Features}<
1	1	{Komprimierter ProductCode}<

Tabelle 1: Aufbau des Darwin Descriptors

Wie bereits bei der komprimierten GUID erläutert, ist der Algorithmus zum Entschlüsseln einer SQUID und somit auch des Darwin Descriptors nicht dokumentiert. Um den Darwin Descriptor in eine lesbare Form umzuwandeln, kann die undokumentierte Funktion `MsiDecomposeDescriptor()` der »Msi.dll« verwendet werden.

## Funktion: MsiDecomposeDescriptor()

Die Funktion `MsiDecomposeDescriptor()` entschlüsselt den Darwin Descriptor und stellt die einzelnen Bestandteile in einer lesbaren und weiter verwendbaren Form dar.

```

UINT MsiDecomposeDescriptor(
    LPCTSTR szDescriptor,
    LPCTSTR szProductCode, // Ermittelter ProductCode
    LPCTSTR szFeatureId, // Ermittelter Name des Fetures
    LPCTSTR szComponentCode, // Ermittelte ComponentId
    DWORD *pcchArgsOffset // Offset der Argumente im Descriptor
);

```

**szDescriptor:** Der Darwin Descriptor, der entschlüsselt werden soll.

**szProductCode:** Zeiger auf eine Variable, die den ermittelten ProductCode aufnehmen soll. Der Puffer muss groß genug sein, um 39 Zeichen aufnehmen zu können. Der Parameter kann Null sein.

**szFeatureId:** Zeiger auf eine Variable, die den ermittelten Namen des Features aufnehmen soll. Der Puffer muss groß genug sein, um 39 Zeichen aufnehmen zu können. Der Parameter kann Null sein.

**szComponentCode:** Zeiger auf eine Variable, die die ermittelte ComponentId aufnehmen soll. Der Puffer muss groß genug sein, um 39 Zeichen aufnehmen zu können. Der Parameter kann Null sein.

**pcchArgsOffset:** Zeiger auf eine Variable zur Aufnahme des Offsets bis zum Beginn der Argumente. Einem Darwin Descriptor können zusätzliche Argumente angefügt werden. Ist der Darwin Descriptor beispielsweise 72 Zeichen lang, wird in dieser Variablen der Wert 73 zurückgegeben, da die Argumente an dieser Position beginnen. Dieser Parameter kommt u.a. bei der Verwendung von qualifizierten Komponenten zum Einsatz. Der Parameter kann Null sein.

Wird die Funktion MsiDecomposeDescriptor() erfolgreich ausgeführt, wird ERROR\_SUCCESS zurückgegeben. Tritt ein Fehler auf, gibt die Funktion ERROR\_INVALID\_PARAMETER zurück. Die Funktion kann ebenfalls verwendet werden, um eine SQUID in eine GUID umzuwandeln. Handelt es sich bei der SQUID um den ProductCode eines installierten Produktes wird ERROR\_SUCCESS zurückgegeben. Handelt es sich um keinen ProductCode, wird zwar ERROR\_INVALID\_PARAMETER zurückgegeben, jedoch in der Variablen szProductCode die umgewandelte GUID abgelegt.