

ORGANISIEREN SIE IHRE ARBEIT MIT ABAPGIT

Inspiricon 

Andra Atanasoae

INSPIRICON AG



Inhaltsverzeichnis

1. Was ist abapGit und wie funktioniert es?	2
2. Vier Schritte bis zur abapGit Erfahrung	3
Schritt 1: Einrichtung des abapGit Servers	3
Schritt 2: GitHub Root Zertifikate herunterladen	3
Schritt 3: SSL-Einrichtung mit dem SAP Trust Manager	4
Schritt 4: Verbindung zu den Git Repositories	7
3. Amazon Web Services	8
4. Online und Offline?	8
5. Fazit	9



1. Was ist abapGit und wie funktioniert es?

Wenn Sie bereits mit der ABAP-Programmierung vertraut sind, aber noch nichts von abapGit gehört haben, stellen wir Ihnen hier zunächst die **Git-Welt** vor.

Git ist ein Versionskontrollsystem mit dem man alle Änderungen einer Datei während ihrer Laufzeit nachverfolgen kann.

Aber was ist ein Versionskontrollsystem? [Ein Auszug aus dem Buch „Pro Git“ von Scott Chacon und Ben Straub schafft Klarheit:](#)

Was ist eine „Versionskontrolle“ und warum sollte ich mich damit beschäftigen? Versionskontrolle ist ein System, das Änderungen an einer oder mehreren Dateien registriert, um spezifische Versionen später noch abrufen zu können. Für die Beispiele in diesem Buch werden wir Software-Quellcodes verwenden, um die Änderungen nachzuverfolgen. In Wirklichkeit könnt ihr das jedoch mit nahezu jeder Datei auf einem Computer machen.

Falls Sie ein Grafik- oder Webdesigner sind und jede Version eines Bildes oder eines Layouts behalten möchten, dann ist ein Versionskontrollsystem (VKS) die richtige Wahl. Damit können Sie eine Datei in einen früheren Zustand zurückversetzen, ein ganzes Projekt rückgängig machen, Änderungen zu verschiedenen Zeitpunkten vergleichen, sehen wer als letztes etwas geändert hat, das möglicherweise Probleme machen könnte, wer einen Aspekt hinzugefügt hat und vieles mehr. Wenn Sie ein VKS nutzen heißt das auch, dass es einfach ist, die Dateien wiederherzustellen, falls Sie etwas durcheinander gebracht oder Dateien verloren haben. Dies alles bekommen Sie für sehr geringe Kosten.

ABAP selbst liefert ebenfalls ein Versionskontrollsystem, aber dieses System gilt nur für die eigene geschlossene Arbeitsumgebung. Es ist also nicht möglich mit „Outsidern“ zusammen zu arbeiten. Code-Sharing ist immer schwierig, besonders jedoch bei Opensource Projekten.

abapGit, der GitClient in ABAP, ist ein Opensource Projekt, das von [Lars Hvam](#) unter Anwendung einer MIT Lizenz initiiert wurde.

Mehr über das abapGit selbst und andere großartige ABAP Opensource Projekte finden Sie in [diesem Blog von Graham Robinson](#), auf dem auch dieser Leitfaden basiert.

Wir konzentrieren uns im Folgenden auf die Installation und Nutzung des abapGit mittels GitHub und Amazon Web Services (AWS) CodeCommit.

In den folgenden Verfahren wurde SAP NetWeaver 7.5 als Arbeitsumgebung verwendet.



2. Vier Schritte bis zur abapGit Erfahrung

Um die Funktionen von abapGit optimal nutzen zu können, sind vier Schritte zur Einrichtung und Anbindung an die Entwicklungsplattform und die Hosting-Dienste notwendig.

Schritt 1: Einrichtung des abapGit Servers

Als erstes müssen wir das abapGit-Projekt auf unserem ABAP-Entwicklungssystem laufen lassen. Wenn Sie den SAP NetWeaver bereits auf deinem Rechner installiert haben, öffnen Sie nun den SAP GUI-Client und erstellen einen ABAP-Report „ZABAPGIT“ über die Transaktionen SE38 oder SE80. Dann wiederholen Sie den Code auch im untenstehenden Link:

<https://raw.githubusercontent.com/abapGit/build/master/zabapgit.abap>

Aktivieren Sie den Bericht und führen ihn aus. Danach wird die abapGit Startseite auf dem Bildschirm angezeigt:



Bitte beachten Sie, dass, wenn Sie dein abapGit-Projekt auf eine neuere Version aktualisieren möchten, Sie nur den Code im Bericht durch die neuesten Einstellungen ersetzen müssen – damit ist das Upgrade schon beendet.

Anders gesagt ist dieses Verfahren ein unkomplizierter Weg, über den Sie jederzeit die neueste Version des abapGit herunterladen können.

Schritt 2: GitHub Root Zertifikate herunterladen

In diesem nächsten Schritt werden wir mit der SSL-Einrichtung fortfahren, die es ermöglicht, die Online-Funktion des Git-Servers zu nutzen.

Im Gegensatz dazu arbeiten Offline-Projekte hinter Firewalls und ohne SSL. Ein SSL, „Secure Sockets Layer“ Zertifikat ist eine Textdatei mit verschlüsselten Daten, die wir auf unserem Server



installieren sollten, um eine sichere, zuverlässige und reaktionsschnelle Kommunikation mit einem Webbrowser sicherzustellen.

Heutzutage ist GitHub die beliebteste webbasierte Gemeinschaft, die von Git organisiert wird. Um die Kommunikation zwischen abapGit und GitHub zu ermöglichen, ist es notwendig, einige Root-Zertifikate auf Ihrem ABAP-System zu installieren. Je nach verwendetem Webbrowser hat man verschiedene Möglichkeiten, die Dateien herunterzuladen, ansonsten kann man sie manuell herunterladen. Der Zweck dieser Root-Zertifikate ist es, die Integrität und Geheimhaltung der Kommunikationsdaten zwischen dem Server und Webbrowser zu sichern.

In unserer Anleitung haben wir Google Chrome als Standard-Webbrowser verwendet. Führen Sie die folgenden Schritte aus, um an die erforderlichen Zertifikate zu kommen:

- Gehen Sie auf die Seite <https://github.com>.
- Klicken Sie auf das Schließfachsymbol neben der Adressleiste und wählen dann „Certificate“.
- Gehen Sie auf die Registerkarte „Details“ und klicken auf „Copy to file...“, um das Zertifikat in eine CER-Datei zu exportieren.
- Geben Sie den Namen der Datei an, wählen Sie den Pfad, in dem die Datei gespeichert werden soll und drücken Sie auf „Finish“.
- Wählen Sie die Registerkarte „Certification Path“ und wiederholen Sie die beiden vorherigen Schritte vom übergeordneten Knoten bis zum Wurzelknoten.

Wenn Sie die Root-Zertifikate manuell herunterladen möchten, navigieren Sie zur GitHub-Plattform und finden heraus, welches Zertifikat sie verwendet; so wie bereits oben gezeigt. Laden Sie diese Zertifikate unter <https://www.digicert.com/digicert-root-certificates.htm> im Bereich „Root Certificates“ herunter.

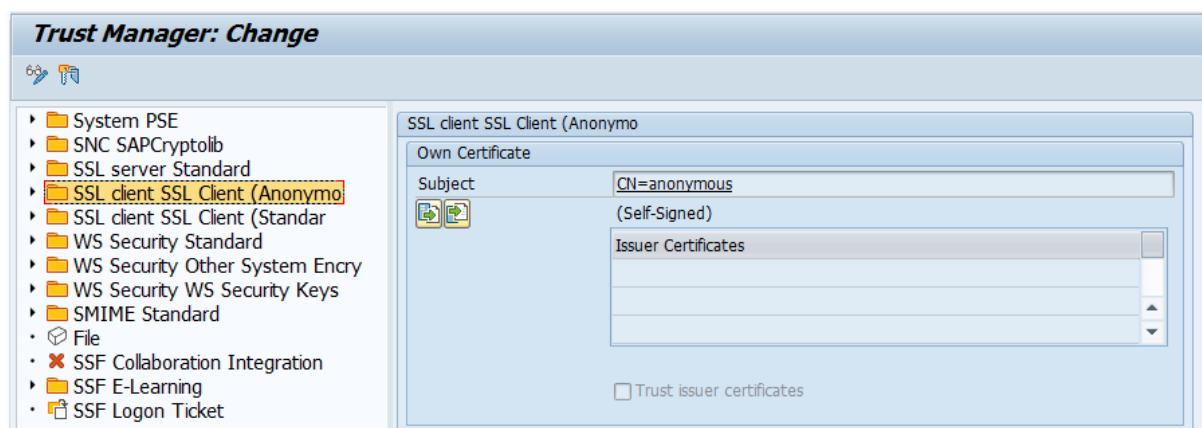
Weitere Informationen finden Sie auch unter <http://docs.abapgit.org/>.

Wir haben nun die Dateien im gewünschten Format und können so die Verbindung zum SAP-System herstellen.

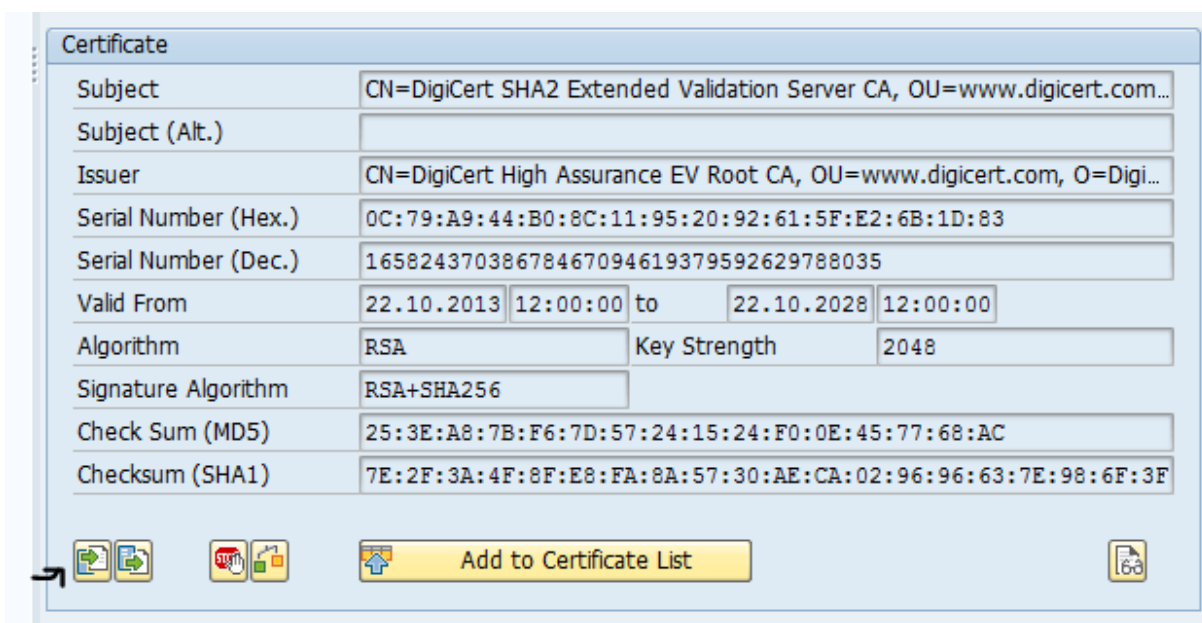
Schritt 3: SSL-Einrichtung mit dem SAP Trust Manager

Damit die zuvor heruntergeladenen Zertifikate bestätigt werden können, müssen wir nun die sogenannte Trust Manager Anwendung im SAP-System installieren. Melden Sie sich dazu an Ihrem SAP-GUI-System an und führen die Transaktion STRUST aus.

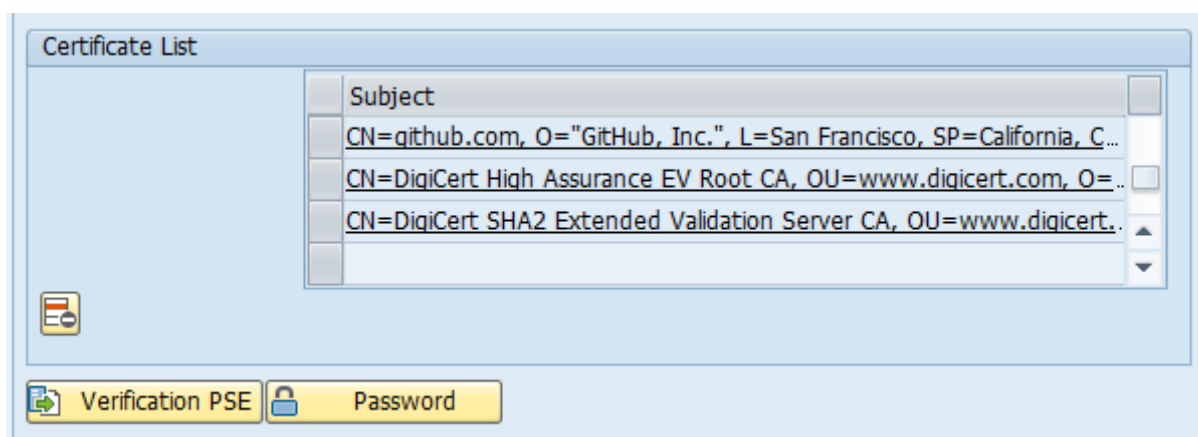
Wechseln Sie zum „Change“ Modus und öffnen Sie aus der linken oberen Objektliste den Ordner „SSL System Client SSL Client (Anonymous)“.



Wählen Sie „Import certificate“ unten links im Feld „Certificate“ aus und fügen Sie die Dateipfade der zuvor aus dem Webbrowser exportierten Zertifizierungsdateien ein.



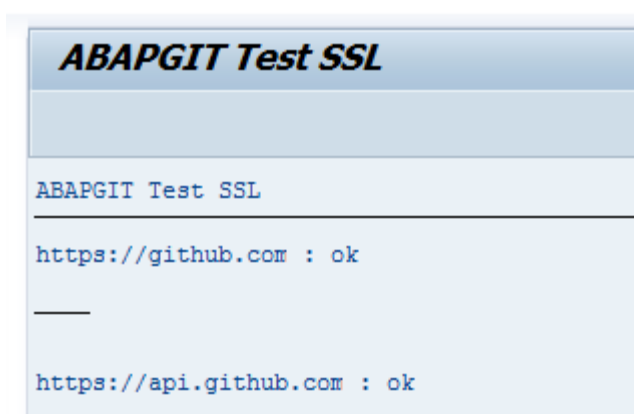
Wählen Sie „Add to Certificate List“ für jede Zertifizierungsdatei einzeln aus, eine nach der anderen. Am Ende sollte das Feld „Certificate List“ wie folgt aussehen:



Speichern Sie die vorgenommenen Änderungen. Die Root-Zertifikate sind nun auf dem SAP-Entwicklungssystem installiert.

Um zu prüfen, ob die ABAP-Werkzeuge mit dem Git-Server kommunizieren können, ist es zwingend notwendig zu testen, ob die Verbindung zwischen ihnen funktioniert. Erstellen Sie einen ABAP-Report „ZABAPGIT_TEST_SSL“ über die Transaktionen SE38 oder SE80, um dies zu veranschaulichen und kopieren Sie den Code aus dem folgenden Link <http://docs.abapgit.org/other-test-ssl.html>

Aktivieren Sie Ihr Programm und führen es aus. Wenn die Verbindung erfolgreich hergestellt wurde, wird folgende Ausgabe erwartet:



Wenn die Verbindung jedoch nicht korrekt funktioniert, achten Sie darauf, dass Sie eventuell zwei Profilparameter in der Transaktion RZ10 einstellen müssen (siehe auch SAP-Hinweis 510007 – Einrichten von SSL auf dem Application Server ABAP, Schritt 7).

In unserem System mussten zum Beispiel folgende Werte eingestellt werden:

```
ssl/ciphersuites = 135:PFS:HIGH::EC_P256:EC_HIGH
```

```
ssl/client_ciphersuites=150:PFS:HIGH::EC_P256:EC_HIGH
```



Bitte beachten Sie, dass Sie den Applikationsserver neu starten müssen oder zumindest in der Transaktion ICM Monitor, SMICM, um den icman-Prozess neu zu starten!

Hier können Sie tiefer in das Thema eintauchen: <https://blogs.sap.com/2008/10/31/calling-webservices-from-abap-via-httpsssl-with-pfx-certificates/>.

Schritt 4: Verbindung zu den Git Repositories

Halten Sie Ihren installierten abapGit-Code auf dem neuesten Stand und erweitern Sie die unterstützten Objekttypen durch das Herunterladen der benötigten Plugins.

Wenn abapGit zum ersten Mal ausgeführt wird, wird die abapGit-Tutorial-Seite angezeigt, wie schon in Schritt 1 gezeigt. Auf der Unterseite der Seite finden Sie den Abschnitt „abapGit related repositories“.

abapGit related repositories

- **install abapGit repo** - To keep abapGit up-to-date (or also to contribute) you need to install it as a repository.
- **install abapGit plugins** - you can also install plugins to extend supported object types

abapGit
1.69.0

Klicken Sie auf „install abapGit repo“, um den Prozess zu starten. Klicken Sie auf „Continue“, um die aktuelle Version von abapGit in das Paket \$ABAPGIT herunterzuladen, das in diesem Fall ein lokales Paket ist. Wählen Sie „Overwrite“, um das zuvor erstellte Programm „ZABAPGIT“ zu aktualisieren und alle abapGit-Werkzeuge zu aktivieren.

Gehen Sie genauso für „install abapGit plugins“ vor. Dann sind Sie bestens dafür vorbereitet, die abapGit-Funktionen auszuprobieren.

abapGit related repositories

- **abapGit installed in package \$ABAPGIT**
- **abapGit plugins installed in package \$ABAPGIT_PLUGINS**

Wichtig dabei ist, dass nicht nur die GitHub-Plattform – ein webbasierter gehosteter Dienst zur Versionskontrolle – mit dem abapGit verbunden werden kann, sondern auch die Möglichkeit der Kommunikation über die Amazon Web Services besteht.



3. Amazon Web Services

Wir bei Inspiricon arbeiten viel mit den Amazon Web Services, daher lag es für uns nahe, den „AWS Service CodeCommit“ als Git-Server für unsere Repositories zu nutzen. Wenn dies auch Ihre Wahl sein sollte, müssen Sie Ihr System nach der Installation zunächst mit den Amazon Web Services verbinden, damit Sie die Online-Funktion der abapGit-Plattform nutzen können.

Amazon Web Services, kurz „AWS“ genannt, bieten zuverlässige, skalierbare und sichere Cloud-Computing-Services, Datenbankspeicher und andere Funktionalitäten zur Unterstützung der Geschäftsprozesse. Um sich mit dem AWS Server zu verbinden, müssen Sie mehrere Schritte ausführen.

Die ersten Schritte sind ähnlich wie die für die Verbindung mit dem GitHub-Server, einschließlich des Herunter- und Hochladens der Root-Zertifikate, wie in Schritt 2 und Schritt 3 beschrieben. Als nächstes müssen IAM-Benutzer im AWS angelegt werden. Es ist wichtig, dass Sie im IAM-Benutzerprofil „HTTPS Git credentials for AWS CodeCommit“ arbeiten.

Als Rechte sollten Sie mindestens die folgenden Berechtigungen vergeben:

- AWSCodeCommitPowerUser AWS Managed Policy
- IAMReadOnlyAccess AWS Managed Policy
- IAMSelfManageServiceSpecificCredentials AWS Managed Policy

Weitere Informationen finden Sie auch unter

<https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html>.

Nach der Erstkonfiguration können Sie im AWS CodeCommit Dashboard eigene Repositories anlegen.

4. Online und Offline?

Das abapGit-Projekt kann in der SAP NetWeaver-Plattform über die Transaktion „ZABAPGIT“ gestartet werden. abapGit verfügt über Online- und Offline-Funktionen.

Die Offline-Funktion bezieht sich einerseits auf die Replikation eines lokalen ABAP-Pakets aus Ihrem System in den abapGit-Workspace. Andererseits erlaubt uns die Online-Funktion, mit Repositories zu arbeiten. Was ist also ein Repository?

Ein Repository, häufig als „repo“ abgekürzt, ist ein Ort, an dem alle Dateien für ein einzelnes Projekt gespeichert werden. Jedes Projekt hat sein eigenes Repository und kann über eine eindeutige HTTPS-Url angesprochen werden. Wir müssen dieses Repository mit einem bestehenden ABAP-Paket aus unserem System verbinden oder ein neues anlegen.



5. Fazit

Immer mehr Unternehmen und Entwickler auf der ganzen Welt haben sich entschieden, Git als zentrale Steuerungssoftware einzusetzen. In der Landschaft der SAP-Branche kommt abapGit den Entwicklern in Form eines in ABAP geschriebenen ABAP-Clients zu Hilfe, der von den Git-Funktionalitäten profitiert.

abapGit ist kostenlos, schnell und bietet ein konsistentes, skalierbares Modell. Ein weiterer großer Vorteil ist, dass abapGit automatisch geänderte Objekte in das remote Git-Repository schiebt und gleichzeitig eine Umgebung ist, die Parallelentwicklungen erleichtert. Es eignet sich optimal für die gemeinsame Nutzung von Code und für die Verfolgung von Änderungen im Entwicklungsprozess.

Wenn Sie eine zentrale Stelle für die Organisation des Quellcodes in Ihren Projekten benötigen, ist abapGit das perfekte Werkzeug.

Sind Sie motiviert und möchten mehr über das Thema erfahren? Gerne informieren wir Sie über die Updates und stellen Ihnen weitere interessante Themen und Projekte vor, an denen wir ebenfalls beteiligt sind. Bleiben Sie mit uns in Kontakt – folgen Sie uns auf [LinkedIn](#), [XING](#) oder [facebook!](#)

Kontakt:

Inspiricon AG
Otto-Lilienthal-Str. 36
71034 Böblingen

www.inspiricon.de

Autor und Ansprechpartner

Andra Atanasoaie Associate
Telefon 7031 714 660 0