

PHP COMmuniziert mit Gupta

PHP ist toll. Kein Zweifel, die Sprache PHP ist neu und hat Zukunft. Demgegenüber ist der Gupta Team Developer (GTD) schon sehr lange am Markt. Insbesondere als Abteilungssoftware hat er sich bewährt. Eine Menge spezieller Team Developer Lösungen laufen still und heimlich im Hintergrund. Wäre es da nicht sinnvoll die vorhandene Businesslogik mit PHP nutzen zu können, ohne das Rad neu erfinden zu müssen?

von Thomas Wiedmann

Vor gar nicht so langer Zeit war 4GL (Programmiersprache der 4. Generation) ein sogenanntes Buzzword in der IT-Szene. Der Gupta Team Developer ist eine 4GL Sprache und als solche gut für eine schnelle Softwareentwicklung. Die Firma Gupta, die damals Centura hieß, hatte während der aufkommenden Interneteuphorie versucht, die vorhandenen SQL-Windows Programme mit einem vorgeschalteten Manager (wer erinnert sich noch an Foresite?) oder später mittels *WebApplication Console* webfähig zu machen. Diese Strategie hat nicht den gewünschten Erfolg gebracht. Deshalb möchte ich hier eine andere Lösung vorschlagen und zwar die Modularisierung der vorhandenen Software in Komponenten. Die inzwischen häufig eingesetzte Technologie COM (Component Object Modell) von Microsoft kann hier eine Lösung bieten, um die vorhandenen Programme in die Zukunft zu retten. Der Team Developer beherrscht seit der Version 2.0 auch COM. Die WebSprache PHP ist ebenfalls unter Windows lauffähig und die COM-Fähigkeiten sind mit der Version 5.0 vollständig überarbeitet worden. Die Geschäftslogik verbleibt in Form von Klassen und Objekten im Gupta (COM-Server) und die GUI wird mit Hilfe von HTML, CSS und PHP in den Browser gebracht. In der Übersicht sieht das Ganze so aus (Bild 1).

Übersicht zum Datenfluß zwischen Browser und COM-Server

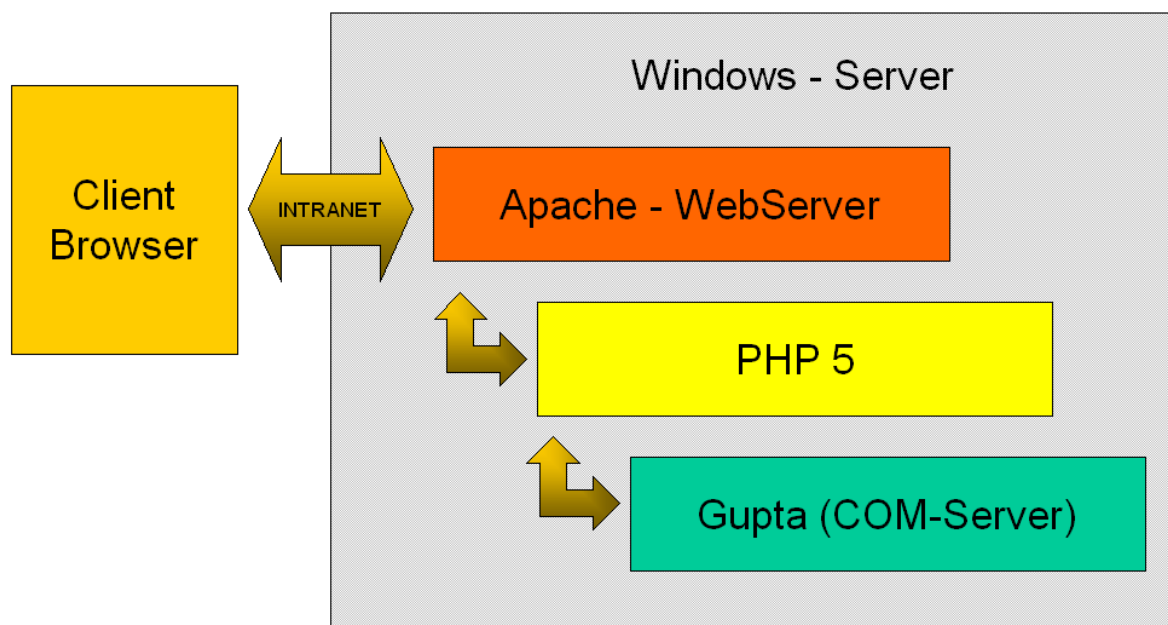


Bild 1: Übersicht über den Datenfluß zwischen Client und Gupta COM-Server

Grundlagen

Bevor wir so richtig loslegen, möchte ich noch schnell die notwendigen Module vorstellen, auf denen der Artikel und die Beispiele basieren.

- Windows 2000
- Apache 1.3.8
- PHP 5.0.1
- Gupta Team Developer v2.1

Wie Sie sehen, es sind nicht die allerneuesten Versionen der jeweiligen Software nötig, um das angedachte Konzept in die Realität umzusetzen. Um ans Ziel zu kommen sind erst einmal keine Investitionen nötig. Also keine schlechte Voraussetzung für einen Testballon.

COM together – Hello World

Beginnen möchte ich mit dem Gupta Team Developer. Um einen ersten Eindruck von der Technik zu bekommen, geht es mit dem berühmten *Hello World* - Beispiel los. Um einen Gupta COM Server zu erstellen, wählen Sie unter den *Wizards* den *COM Class Wizzard* als *CoClass* Komponente aus. Als Name der neuen *CoClass* empfiehlt sich *Application* (Bild 2).

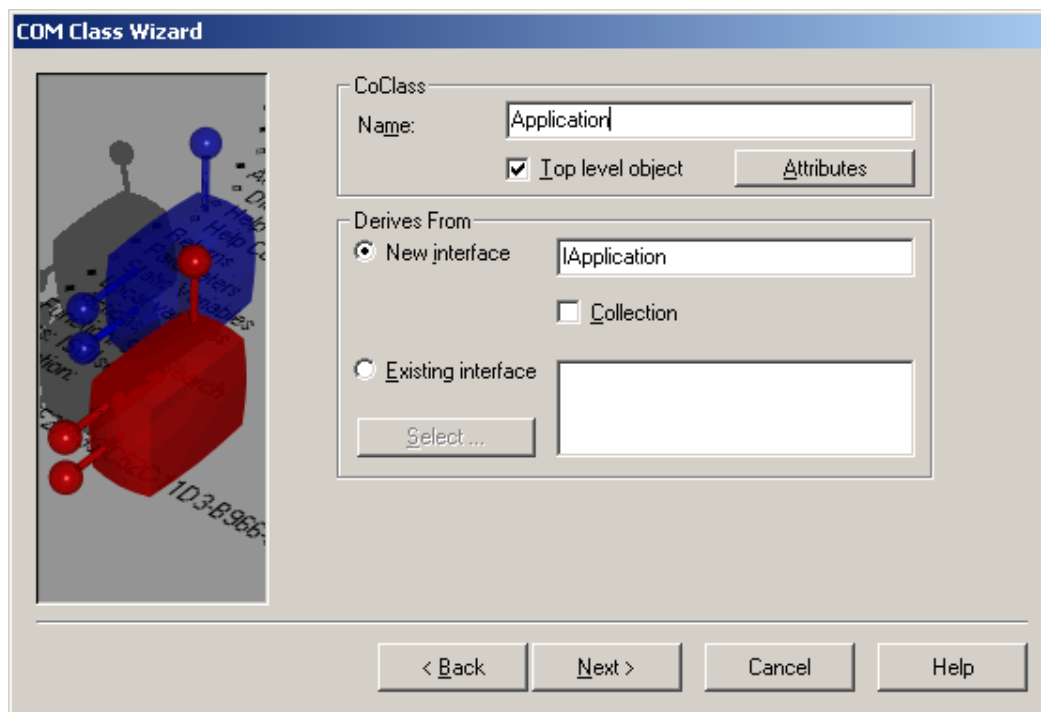


Bild 2: Mit Hilfe des Gupta COM Class Wizard eine neue CoClass erzeugen

Aus der CoClass mit dem Namen *Application* liefert der Wizard zusätzlich ein sogenanntes Interface *IApplication*. Wie Sie später sehen können, entspricht so ein Interface nahezu den bekannten *Functional Classes*. In der Terminologie spricht man hier auch von Methoden. Anschließend geht es mit [Next] weiter.

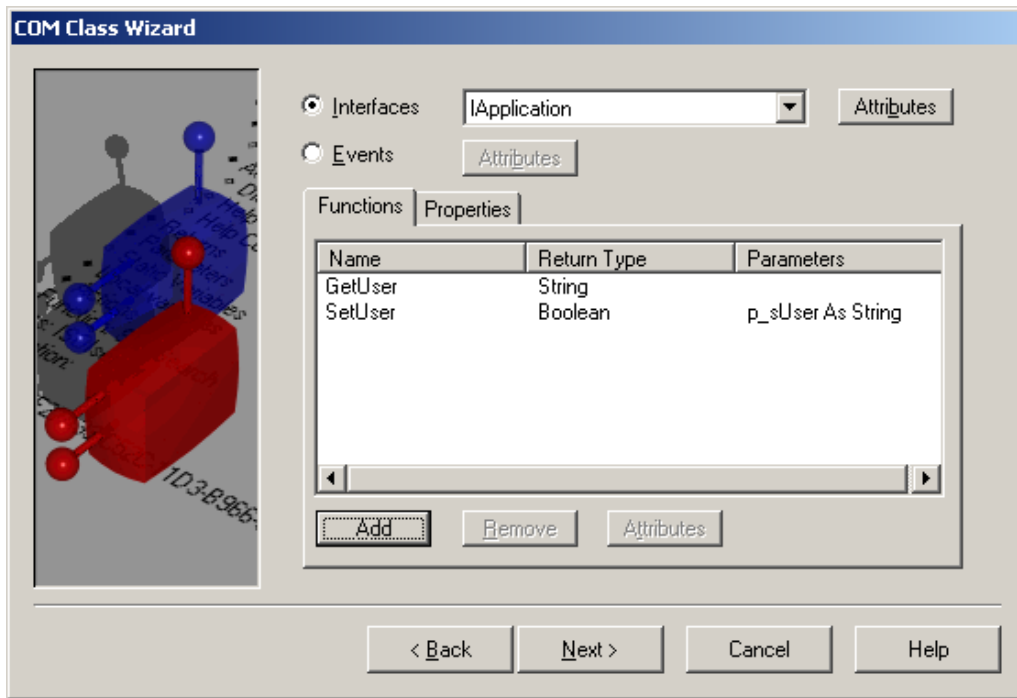


Bild 3: Zwei Funktionen (Set/Get) sind hinzugekommen.

Mittels [Add] sind zwei Funktionen oder Methoden hinzugekommen. Das Property *User* kann nun der CoClass übergeben und abgeholt werden. Alternativ dazu können auch per Karteireiter *Properties* solche Set/Get Methoden (Funktionen) erzeugt werden. Als Methodenname wird dann *PropGet-* bzw. *PropSetUser* mit einer passenden Instanzvariablen erzeugt. Welche Methode Sie wählen bleibt Ihnen überlassen, aber generell können Sie davon ausgehen, dass Sie nicht alle Properties in dem Wizard anlegen werden, sondern später den größten Teil direkt im GTD kodieren werden. Der *COM Class Wizard* soll nur die Möglichkeiten aufzeigen. Jetzt noch [Next] und [Finish] anklicken und fertig ist die erste *CoClass* (Bild 4).

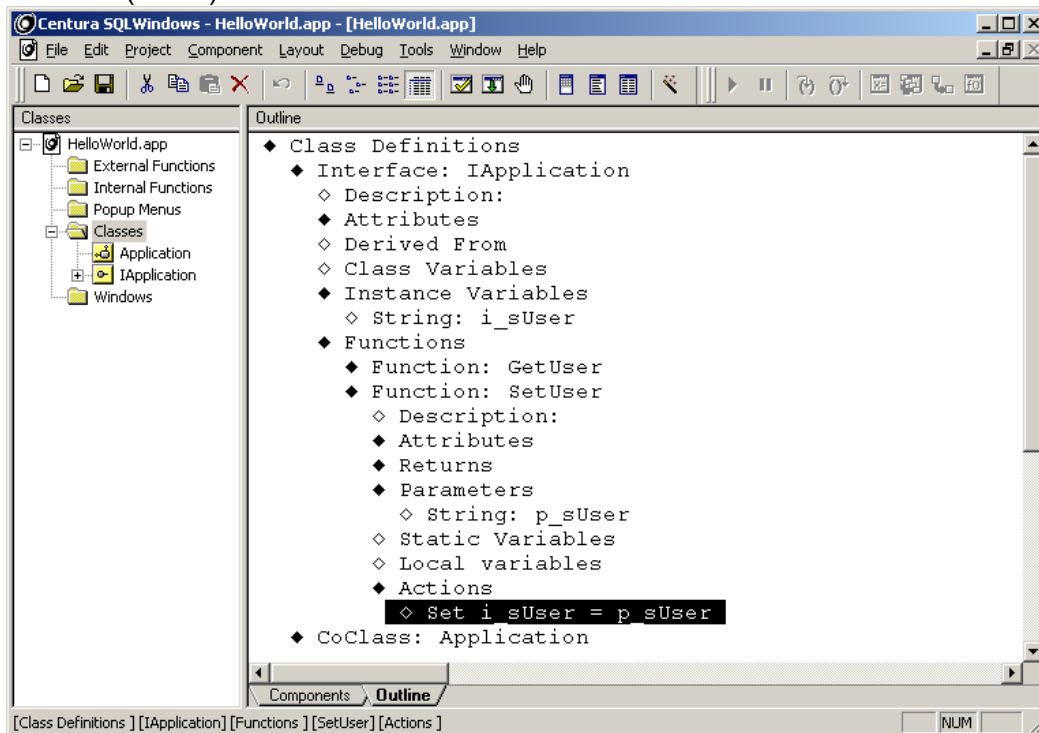


Bild 4: Der generierte Code der neuen CoClass von HelloWorld.app
Manuell ergänzt habe ich jetzt noch die Instanz Variable *i_sUser* sowie die jeweiligen

Zuweisungen in den Funktionen *Set-* bzw. *GetUser*. Damit ist der erste Gupta COM Server auch schon fertig. Im nächsten Schritt muß er nun Windows bekannt gemacht, also registriert werden. Dazu muß zuerst das Programm erzeugt werden. Unter dem Menüpunkt *Project / Build Settings ..* (Bild 5) muß noch die Einstellung *Target Type* angepaßt werden, damit nicht ein Standard EXE, sondern ein sogenannter *Local COM Server (EXE)* erstellt wird.

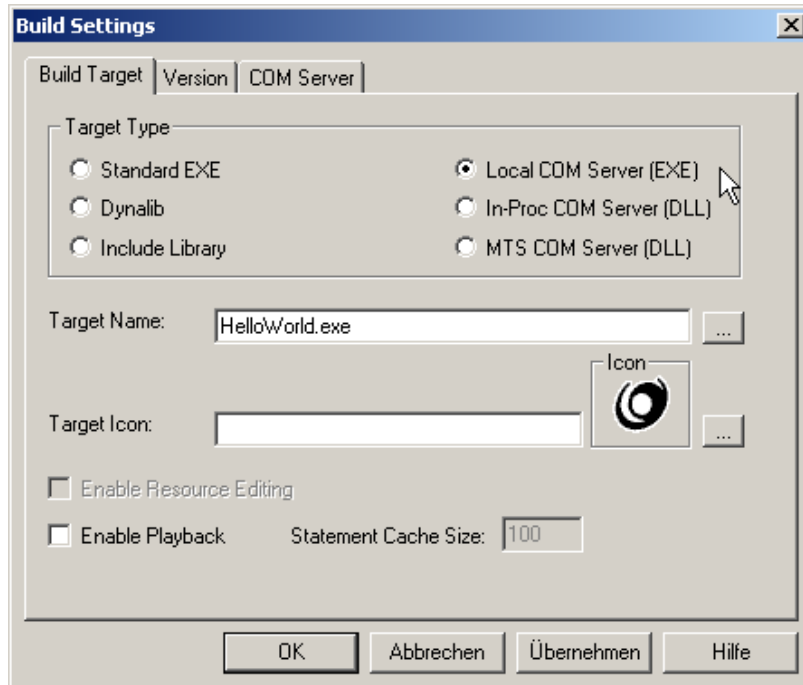


Bild 5: Build Settings einstellen auf - Local COM Server (EXE)

Sie sehen, es gibt mehrere Möglichkeiten. Ein *Local COM Server (EXE)* läuft in einem eigenen Adressraum und stört im Fehlerfall keine andere Applikation auf dem Server. Die Variante *In-Proc COM Server (DLL)* läuft im selben Adressraum, wie das aufrufende Programm (PHP bzw. Apache). Die sichere Lösung ist somit der *Local COM Server (EXE)*. Nun kann über *Project / Build: HelloWorld.exe* das Programm erzeugt werden. War dies erfolgreich, fehlt noch die eingangs besprochene Registrierung. Dies wird einfach mit dem Menüpunkt *Project / Register Server* erledigt. Sehen Sie jetzt noch die Bestätigung: *Component successfully registered* dann haben Sie es geschafft. Der neue COM Server ist registriert.

PHP gibt das Kommando..

Nun ist es an der Zeit, sich um die PHP Seite zu kümmern. Für das erste Beispiel ist kein Apache-WebServer notwendig. Das PHP-Skript wird direkt in der Kommandozeile ausgeführt. Wenn PHP lauffähig ist und der Gupta COM-Server korrekt registriert ist, sollte das Ergebnis wie folgt aussehen. Übrigens, für COM benötigen Sie keine zusätzlichen PHP Extension in der *php.ini* einrichten. Unter Windows ist COM ein Teil des sogenannten PHP core.

Ausführen auf Kommandoebene:

```
D:\Test>php HelloWorld.php
Hallo, PHP ist willkommen<br>
D:\Test>
```

Das PHP-Skript *HelloWorld.php* wird dem Interpreter (php.exe) als Parameter übergeben und ausgeführt. Das Skript übergibt dem COM.-Server den Text »PHP« (Set) und holt den Wert (Get) wieder ab. Anschließend wird das Ganze per *echo* angezeigt.

Das komplette Skript sieht so aus:

```
HelloWorld.php:
<?php
/**
 * PHP COMmuniziert mit Gupta
 *
 * HelloWorld.php PHP 5.0.1
 * (c) 18.04.2005 Thomas Wiedmann http://www.twiedmann.de
 */
$user = '';

/**
 * ein Gupta Objekt erzeugen
 */
$oGupta = new COM("HelloWorld.Application");

/**
 * Wert an den Server übergeben und wieder abholen
 */
$oGupta->SetUser('PHP');
$user = $oGupta->GetUser();
echo 'Hallo, '.$user.' ist willkommen<br>';

/**
 * zum guten Schluß aufräumen
 */
$oGupta = null;
unset($oGupta);
?>
```

Der interessanteste Teil ist der Befehl `$oGupta = new COM("HelloWorld.Application")`. Damit wird eine neue Instanzvariable `$oGupta` des Gupta COM-Servers *HelloWorld.Application* erzeugt. Damit haben wir nun Zugriff auf alle exportierten Methoden. Beispielsweise wird per `$oGupta->SetUser('PHP')` der Wert »PHP« übergeben und anschließend wieder abgeholt. Der Inhalt der Variable wird schließlich ausgegeben. Klappt alles wie erhofft, entsteht der schöne Satz »Hallo, PHP ist willkommen«. Ist die Party vorbei, wird aufgeräumt. So auch hier. Es genügt (ab PHP 5) die Instanzvariable `$oGupta` auf *null* zu setzen und mit `unset()` freizugeben. Sie sehen, das komplette Beispiel ist sehr einfach gehalten, enthält aber alle wesentlichen Teile, um weitere Versuche zu starten.

Feiertagsberechnung – Wann ist Ostern?

Als Basis für den nächsten Schritt eignet sich ein älteres Beispiel (GetFeiertage.zip [1]) von meiner Centura Downloadseite, das ich vor einigen Jahren erstellt habe. Da voraussichtlich auch in den nächsten Jahren Ostern nicht abgeschafft werden wird, lohnt es sich die Funktion als Grundlage für eine schicke Kalenderfunktion. Zum Glück habe ich früher die Feiertagsberechnung [1] bereits als *Functional Class* konzipiert, somit ist eine weitergehende Nutzung sehr einfach möglich. Um den Programmcode innerhalb eines COM-Servers zu nutzen, brauchte ich nur die Klasse per Copy/Paste in das neue Programm *PHP_Gupta_COMmunication.apl* zu kopieren (Bild 6), das ich ebenfalls – analog dem obigen Beispiel - per COM-Class Wizard erstellt habe.

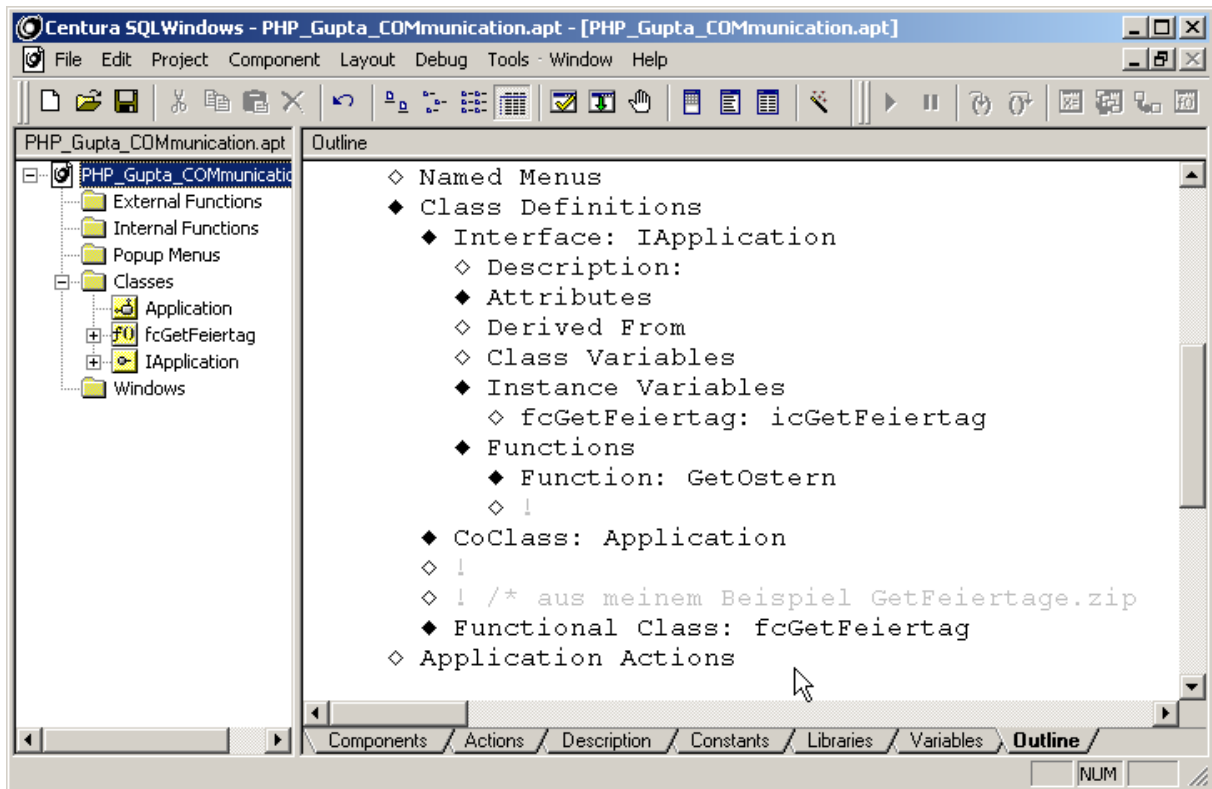


Bild 6: Der Code der Feiertagsberechnung wird als neue Functional Class *fcGetFeiertag* eingefügt.

Folgendes ist jetzt zu beachten: Die Functional Class *fcGetFeiertag* liegt ausserhalb dem Interface *IApplication*, wird also folglich später beim Registrieren nicht veröffentlicht. Dies erfolgt nur bei der Methode oder Funktion *GetOstern*. Warum? Sie gehört zu der übergeordneten CoClass *Application* und nur die dahinter liegenden Methoden werden veröffentlicht. Damit *GetOstern()* Zugriff auf die Feiertagsberechnung bekommt, vererben wir die ganze *Functional Class* in Form der Instance Variable *icGetFeiertag*. War doch ganz einfach, oder? Innerhalb *GetOstern()* ist nur folgender Code notwendig:

```
Function: GetOstern
[... ]
Set ldtOstern = icGetFeiertag.CalcOstern(p_nYear)
Return ldtOstern
```

Es erfolgt eine Parameterübergabe *p_nYear*, welches das gewünschte Jahr für die Osterberechnung enthält. Der Rückgabewert von *CalcOstern()* ist vom Datentyp *Date/Time*. Nun wenden wir uns wieder der PHP Seite zu. Vergessen Sie aber nicht das Gupta-Programm zu erstellen (Build: *PHP_Gupta_COMMunication.exe*) und zu registrieren. Wenn Sie das Programm mehrfach ändern und diverses ausprobieren, muß der COM-Server (das EXE) zuerst per *Un-Register Server* abgemeldet werden, bevor ein neues *Register Server* erfolgen kann.

Eindeutige Dispatch Number

Wenn Sie innerhalb dem *IApplication Interface* »von Hand« neue Funktionen (Methoden) programmieren möchten, müssen Sie auch manuell für eine eindeutige *Dispatch Number* sorgen. Zum Teil ist die GUI bereits so intelligent die *Dispatch Number* für eine neue Funktion eindeutig zu vergeben, aber unter gewissen Umständen klappt dies nicht immer.

```
PHP_Gupta_COMmunication.apt:
```

```
Function: GetOstern  
Description:  
Attributes  
  Dispatch Number: 1  
  Help String:  
  Help Context: 0  
[...]
```

Jede Methode oder *Function* eines Interface trägt eine *Dispatch-ID* – eine Kennziffer, die die Methode eindeutig identifiziert. Die *Dispatch ID* oder - *Number* darf somit nicht doppelt vergeben werden.

PHP feiert Ostern

Im nächsten Schritt geht es wieder um PHP. Diesmal nicht nur um die Kommandozeile, sondern um ein vollständiges Formular zur Abfrage und Anzeige des Ostersonntags. Auch eine Eingabevalidierung wird gezeigt. Genau das, was im harten Praxisbetrieb zwingend notwendig ist (Bild 7). Voraussetzung dafür ist ein lauffähiger Apache Server und PHP läuft als sogenanntes Apache-Modul.

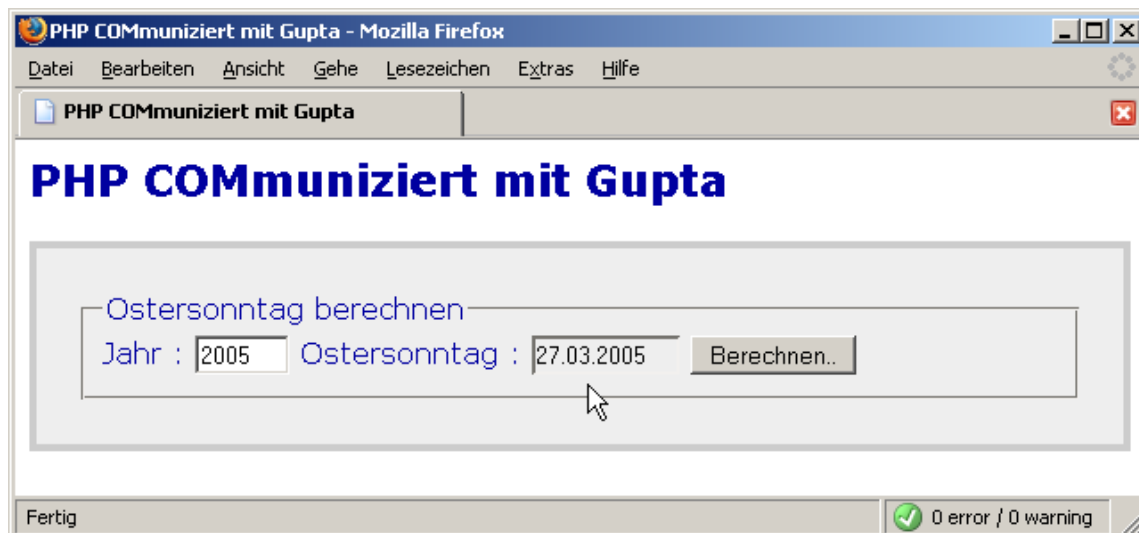


Bild 7: Datenabfrage und Berechnung von Ostern im Web (PHP_Gupta_COMmunication.php)

Mit PHP, etwas HTML und CSS habe ich Ihnen ein schönes Formular zusammen gestellt. Im Eingabefeld »Jahr« wird die Jahreszahl eingegeben, zu dem der Ostersonntag berechnet werden soll. Innerhalb des PHP-Skript wird der Eingabewert ausgewertet und auf seinen Wertebereich geprüft. Das Ausgabefeld »Ostersonntag« ist per HTML Attribut *readonly* für Eingaben gesperrt. Mit etwas CSS wird zudem der Hintergrund – analog des Gupta Befehls *SalDisableWindow ()* – grau dargestellt. Ich denke, damit fühlen Sie sich als Gupta-Programmierer beinahe wie zu Hause.

Formular versenden und Daten auswerten

Bleibt noch der [Berechnen..] Knopf. Wird darauf geklickt, sendet der Browser die Seite zurück an den Server, dort werden die Inhalte geprüft. Anschließend wird die Seite neu zusammengestellt und komplett an den Browser zurück geschickt. Die inhaltliche Prüfung findet somit serverseitig statt. Aber keine Angst, all das geht ziemlich schnell. Denken Sie beispielsweise mal an die Reaktionszeit von Google.

```

PHP_Gupta_COMmunication.php
[...]
/**
 * Werteübergabe und Prüfung nach [Submit] wenn php.ini:
 * register_globals = Off steht, was zu empfehlen ist
 */
if (isset($_POST["jahr"])) { $nYear = (INT) $_POST["jahr"];
  if (($nYear >= 1900) and ($nYear <= 2099)) { /* gültiger Bereich */ }
  else { $nYear = 2005; }
}
if (isset($_POST["ostersonntag"])) { $sOstern = $_POST["ostersonntag"]; }

/**
 * ein Gupta Objekt erzeugen (Instanz erzeugen)
 */
$oGupta = new COM("PHP_Gupta_COMmunication.Application");

/**
 * Wert an den Server übergeben und wieder abholen
 */
$sOstern = $oGupta->GetOstern($nYear);
[...]

```

Die Datenübergabe innerhalb dem PHP-Skript erfolgt mit der vordefinierten Array-Variable `$_POST`. Gefüllt wird `$_POST` aus der sogenannten HTTP_POST Variablen. Die Post Variablen (die Eingabefelder) sind per Formular mit der *action method="post"* an den Server geschickt worden. Noch ein Wort zur Modularität: Die Verarbeitung (das PHP-Skript) und die GUI (das HTML-Formular) ist im Source schön voneinander getrennt, was generell anzustreben ist. Was nun noch fehlt, ist der HTML Teil. Das notwendige Formular baut sich wie folgt auf:

```

PHP_Gupta_COMmunication.php
[...]
<h2>PHP COMmuniziert mit Gupta</h2>
<form method="post"
action="<?php echo $_SERVER['PHP_SELF'] ?>"
  accept-charset="ISO-8859-1, ISO-8859-2">
  <legend>Ostersonntag berechnen</legend>
  Jahr : <input type="text"
    name="jahr"
    size="5"
    maxlength="4"
    class="edit"
    value="<?php echo $nYear ?>" >
  Ostersonntag : <input type="text"
    name="ostersonntag"
    size="10"
    maxlength="10"
    class="disable"
    value="<?php echo $sOstern ?>"
    readonly="readonly">
  <input type="submit"
    name="submit"
    value="Berechnen.." >
</form>
[...]

```

Für all die, die noch kein HTML kennen und können sieht dies bestimmt ziemlich verworren aus, aber keine Angst, bei genauerer Betrachtung klärt sich das eine oder andere. Die fünf wichtigsten Dinge für das Verständnis habe ich fett und unterstrichen markiert.

- `method="post"` Beschreibt die Art, wie die Formulardaten zum Server kommen. POST (im Formular), GET per URL Variable.
- `action="<?php.."` Das Formular wird an „sich selbst“ zum Auswerten verschickt. Es ruft sich quasi selbst auf. Ein eingebettetes PHP-Miniskript liefert dazu den eigenen Skriptnamen des HTML Dokumentes.
- `name="jahr"` Name des Eingabefeldes "Jahr :"

- name="ostersonntag" Name des (readonly) Eingabefeldes ="Ostersonntag :"
- type="submit" Der Sendeknopf "Berechnen.." als Sonderform eines Eingabefeldes

In dieser Technik liegt die ganze Weisheit zum Arbeiten mit Formularen begraben. Wenn Sie diese Grundlagen verstanden haben, ist alles weitere reine Fleißarbeit. Wobei mir klar ist, dass ich Ihnen hier andernfalls einiges an Hausaufgaben aufgehalst habe. Das ist mir zu Beginn auch so gegangen. Aber wie heißt es so schön: Erfolg ist 10% Inspiration und 90% Transpiration.

PHP 5 – COM Exceptions

Mit der Version 5 von PHP haben die Entwickler auch das sogenannte Exception Handling integriert. Mit *try* und *catch* ist es nun möglich, eventuelle »fatale« Fehler von COM abzufangen und einen kontrollierten »Abgang« zu programmieren.

```

PHP_Gupta_COMmunication_try_catch.php:
[...]
/**
 * ein Gupta Objekt erzeugen (instanzieren)
 */
try {

    $oGupta = new COM("PHP_Gupta_COMmunication.test"); // ← Fehler

    /**
     * Wert($nYear) an den Server übergeben und Ergebnis($sOstern) abholen
     */
    $sOstern = $oGupta->GetOstern($nYear);

    /**
     * zum guten Schluß aufräumen
     */
    $oGupta = null;
    unset($oGupta);

} /**
 * COM Exception abfangen und analysieren
 */
catch (com_exception $exception) {
    echo "COM exception ist aufgetreten: file=".$exception->getFile().
        " in Zeile (".$exception->getLine().")<br>\n";
    echo "Fehlernachricht : ".$exception->getMessage()."<br>\n";
    die();
}
[...]

```

Obiges Beispiel kann nicht korrekt ausgeführt werden, da die Methode *...test* nicht existiert. Mittels *catch* wird der Fehler abgefangen und eine ausführliche Fehlermeldung ausgegeben.

```

COM exception ist aufgetreten: file=d:\Test\php_gupta_communication_try_catch.php
in Zeile (25)
Fehlernachricht : Failed to create COM object `PHP_Gupta_COMmunication.test':
Ungültige Syntax

```

Sicherheit und Zugriffsrechte?

Mit dem kleinen aber feinen Programm habe ich gezeigt, wie es möglich ist, die Web-Sprache PHP und die »Rich-Client« Software von Gupta zur Zusammenarbeit zu bewegen. Als Bindeglied dient COM, das Komponentenmodell von Microsoft. Die notwendigen Programme (Apache, PHP, COM-Server) sind auf einem Windows-Server installiert. Um jetzt einem (externen) Client – also dem Browser – die notwendigen Zugriffsrechte zu geben, sind Anpassungen mit dem Programm DCOMCNFG.EXE (auf dem Server) notwendig (Bild 8).

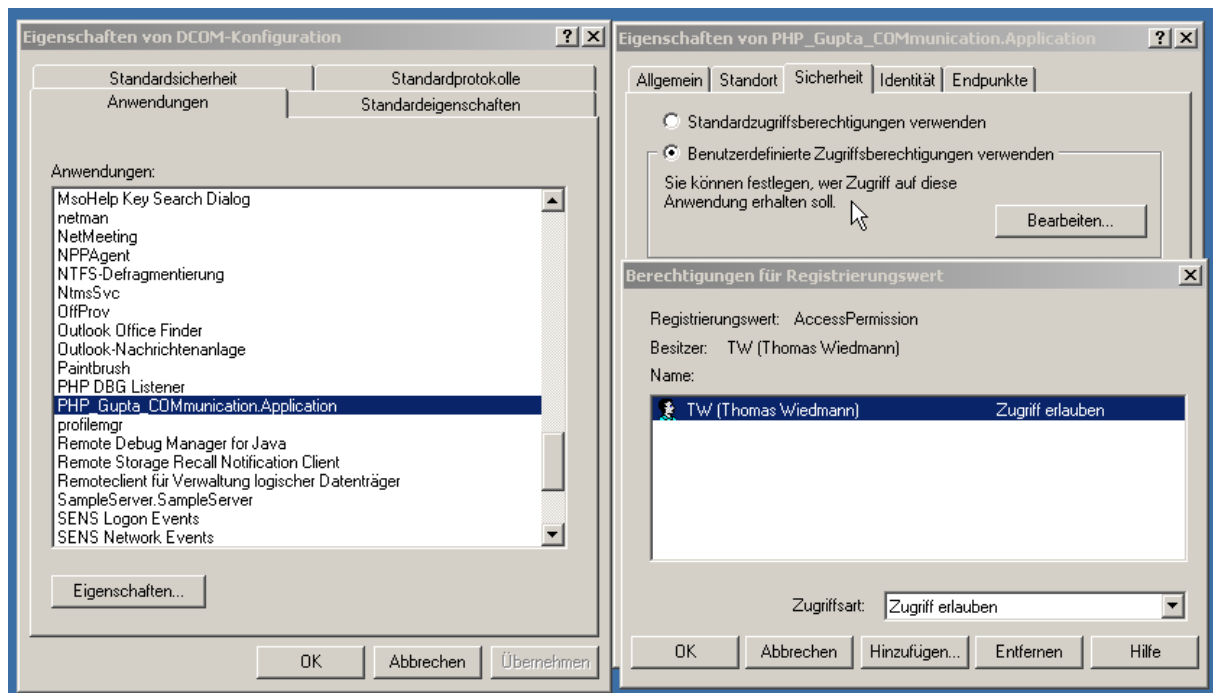


Bild 8: Notwendige Einstellungen bei der Zugriffssicherheit mit DCOMCNFG.EXE

Nachdem DCOMCNFG.EXE gestartet ist, muß auf der Karteikarte »Anwendungen« der Eintrag »PHP_Gupta_COMMunication.Application« gesucht werden. Anschließend auf [Eigenschaften] klicken. Unter »Sicherheit« muß das selbe Benutzerkonto zugewiesen werden, unter dessen Konto auch der Apache-Server läuft. Dies gilt für alle drei Auswahlfelder unter »Sicherheit«. Weiterhin muß unter der Karteikarte »Identität« die Auswahl »Benutzer, der die Anwendung startet« ausgewählt werden. Nach dem Speichern und einem Neustart des Apache sollte der Zugriff nun möglich sein.

Fazit

Anhand des gezeigten Beispiels ist schnell klar geworden, was die Verbindung PHP und Gupta zu leisten in der Lage ist. Mit COM als Klebekitt können Sie – innerhalb der Windowswelt – mit relativ geringem Aufwand bestehende Lösungen nutzen und in Ihr Intranet integrieren. Ich nenne hier bewusst das Intranet, denn der Ausgangspunkt waren ja die mit viel Mühen programmierten Abteilungslösungen.

Literatur

[1]: Feiertagsberechnung: c't 18/1988 S.191

Abkürzungen und Links

- Apache Apache Web-Server (<http://www.apache.org>)
- COM Component Object Model (<http://www.microsoft.com/Com/resources/comdocs.asp>)
- CSS Cascading Stylesheets (<http://www.w3.org>)
- HTML HyperText Markup Language (<http://de.selfhtml.org/>)
- PHP Hypertext Preprocessor (<http://www.php.net>)