

# Typumwandlungsoperator (dateconv1.cpp)

```
#include <string>
#include <ctime>
#include <iostream>
using namespace std;

class Datum {
public:
    Datum(); // Default-Konstruktor
    void aktuell(); // Systemdatum setzen
    operator string() const; // Typumwandlungsoperator
private:
    int tag_, monat_, jahr_;
};

Datum::Datum(){ aktuell(); } // Defaultkonstruktor

void Datum::aktuell() { // Systemdatum eintragen
    time_t now = time(NULL);
    tm *z = localtime(&now); // Zeiger auf struct tm
    jahr_ = z->tm_year + 1900;
    monat_ = z->tm_mon+1; // localtime liefert 0..11
    tag_ = z->tm_mday;
}

// besser: durch Methode string toString() ersetzen!
Datum::operator std::string() const {
    std::string temp("tt.mm.jjjj");
    temp[0] = tag_/10 + '0'; // implizite Umwandlung in char
    temp[1] = tag_%10 + '0';
    temp[3] = monat_/10 + '0';
    temp[4] = monat_%10 + '0';
    int pos = 9; // letzte Jahresziffer
    int j = jahr_;
    while(j > 0) {
        temp[pos] = j % 10 + '0'; // letzte Ziffer
        j = j/10; // letzte Ziffer abtrennen
        --pos;
    }
    return temp;
}

int main() {
    Datum d;
    string s1 = d; // implizit
    cout <<"s1 = "<<s1<<endl;
    string s2 = (string)d; // explizit
    cout<<"s2 = "<<s2<<endl;
    string s3 = static_cast<string>(d); // explizit
    cout<<"s3 = "<<s3<<endl; cin.get();
}

// s1 = 06.05.2010
// s2 = 06.05.2010
// s3 = 06.05.2010
```