

Augel's "Tipps & Tricks" für System i Entwickler

Liebe System i Freunde,

hier eine weitere Ausgabe unseres neuen Projekts speziell für System i Entwickler.

Auch im Kapitel 4 gilt wie immer: "Klein und mit wenig Aufwand große Effekte erzielen."

Einige Möglichkeiten schlummern sozusagen im Verborgenen. Teilweise ist nur das richtige Kombinieren von verschiedenen Funktionen notwendig, um neue Möglichkeiten in eigenen Anwendungen zu realisieren. V6R1 oder V7R1 sind dazu nicht notwendig, da diese Funktionen schon seit Langem möglich sind.

Vielen Dank auch für Ihr fachliches und nützliches Feedback. Ich freue mich besonders, dass viele Kollegen diese interessanten, seit Langem vorhandenen Möglichkeiten so gut gebrauchen können.

Programme und Befehle ferngesteuert ausführen

Steuerung per Message-Break-Handler-Pgm, so wie IBM in OS/400 oder i.

Ich bezeichne bei Schulungen diese Technik gerne als brandneu mit dem Zusatz, dass sie **erst seit 1979** mit dem System/38, dem Vorgänger der AS/400, iSeries, System i, **existiert und allgemein zur Verfügung steht**. Die Struktur dieser Funktion ist "**Objekt basierend**", da ein einzelnes Pgm als eine Art Treiber-Pgm genutzt wird, um von verschiedensten Stellen und Situationen aus, unterschiedlichste feste oder variable Funktionen, Programme und Befehle auszuführen. Die Ausführung kann dabei in jedem angemeldeten Job ausgeführt werden, wo der **Message-Break-Handler** aktiv ist.

Hier einige Beispiele zur Anregung Ihrer Fantasie, welche weiteren Funktionen und Anforderungen Sie damit einfach, leicht und schnell realisieren könnten:

Ein Batch-Pgm kann mit dieser Technik bei Entstehung eines Problems seinen eigenen Joblog an eine bestimmte Person senden, wie eine Durchbruch-Nachricht. Oder ferngesteuert mit einer gesendeten Nachricht wird ein Excel Sheet automatisch bei einem anderen Benutzer geöffnet. Abläufe können einfach und elegant gesteuert und automatisiert werden.

Dieses MsgBrk-Pgm beinhaltet zusätzlich auch das Darstellen eingehender Nachrichten als Status-Nachrichten, welche automatisch in Zeile 24 angezeigt werden.

MSG-Handling ist eine der stärksten Funktionen der Betriebssysteme (CPF, OS/400, i), angefangen mit dem IBM System/38 bis zum heutigen Power System i. Viele interne Funktionen werden mit dieser Technik gesteuert.

Alle BLAUEN Angaben sind feste Werte, die Teil des Schlüsselwortes sind oder zur Steuerung gehören.

Alle ROTEN Angaben sind variable Werte und müssen auf die einzelnen Bedürfnisse angepasst werden.

Nicht vergessen, Ihr Feedback ist wichtig! Senden Sie eine E-Mail, wie Ihnen die "Tipps & Tricks" gefallen, was Sie verbessern würden, oder eigene Tipps aus Ihrer Trickkiste, einfach an: Augel@SSS-Software.de

Teil 1: Setup

CHGMSGQ und MONMSG zum Aktivieren der Message-Break-Handler-Funktion einfach im CL-Start-Pgm im Benutzerprofil eintragen. Zukünftig werden eingehende Nachrichten nicht mehr stumpf auf einem schwarzen Bildschirm ausgegeben, sondern an das angegebene MsgBrk-Pgm übergeben und ausgeführt.

```
****** Datenanfang *******
0001.00
                   PGM
                              MSGID(CPF0000)
0003.00
                   CHGMSGO
                              MSGQ(*USRPRF) DLVRY(*BREAK) +
0004.00
                                PGM(SSS TOOLS/SSS MSGBRK)
                              MSGQ(*WRKSTN) DLVRY(*BREAK)
0005.00
                   CHGMSGQ
0007.00
                   CALL QCMD
0008.00
                   ENDPGM
0009.00
```



Teil 2: Message-Break-Handler-Pgm

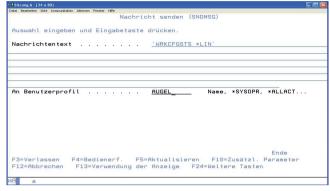
Das MsgBrk-Pgm steuert jetzt nicht nur das elegantere Anzeigen von Nachrichten, sondern ermöglicht jetzt auch über QCMDEXC das Ausführen von Befehlen und Programmen. Wird eine Nachricht an einen Benutzer gesendet, so wird das MsgBrk-Pgm bei diesem Benutzer und in seinem Job ausgeführt, sozusagen ferngesteuert.

```
0001.00 /**************************
            © Copyright SSS/OSI Software Lab. 1998
0004.00
                      PARM(&MSGOUE &LIBARY &MSGKEY)
0005.00
0006.00
                       VAR(&MSGQUE) TYPE(*CHAR) LEN(10)
0007-00
            DCT.
                       VAR(&LIBARY) TYPE(*CHAR) LEN(10)
0008.00
                       VAR(&RTNCDE) TYPE(*CHAR) LEN(01)
            DCT.
                       VAR(&MSGKEY) TYPE(*CHAR) LEN(04)
0009.00
0010.00
                       VAR(&MSG#ID) TYPE(*CHAR) LEN(07)
0011.00
            DCT
                       VAR(&MESSGE) TYPE(*CHAR) LEN(250)
VAR(&BLANKM) TYPE(*CHAR) LEN(250)
0012.00
            DCL
                       VAR(&SENDER) TYPE(*CHAR) LEN(80)
0013.00
0014.00
            DCT.
                       VAR(&MSGTYP) TYPE(*CHAR) LEN(2)
0015-00
            DCT.
                       VAR(&MSGKEY) TYPE(*CHAR) LEN(4)
                       VAR(&JOBNAM) TYPE(*CHAR) LEN(10)
0016.00
            DCT
0017.00
                     VAR(&LENGTH) TYPE(*DEC) LEN(15 5) VALUE(250)
           DCL
0018.00
            MONMSG
                       MSGTD(CPF0000)
                      MSGQ(&LIBARY/&MSGQUE) MSGTYPE(*ANY)
0019.00
       L1: RCVMSG
RMV(*NO)
0020.00
                         KEYVAR(&MSGKEY) MSG(&MESSGE) -
0021.00
                        MSGID(&MSG#ID) SENDER(&SENDER) +
0022.00
                        RTNTYPE(&MSGTYP)
0023.00
            RTVJOBA
                       JOB(&JOBNAM)
0024.00
                      COND(&MESSGE *EQ &BLANKM) THEN(GOTO +
0025.00
                        CMDI-BI-(PGMEND))
0026.00
0027.00
                      COND(%SST(&MESSGE 1 1) *EQ ,/') THEN(DO)
            RMVMSG
0028.00
                       MSGO(&LIBARY/&MSGOUE) MSGKEY(&MSGKEY)
0029.00
                       VAR(&MESSGE) VALUE(%SST(&MESSGE 2 249))
            CHGVAR
                       PGM(QCMDEXC) PARM(&MESSGE &LENGTH)
0030.00
0031.00
            GOTO
                       CMDLBL(L1)
            ENDDO
0032.00
0033.00
            SNDPGMMSG MSGID(CPF9898) MSGF(QCPFMSG)
0034.00
MSGDTA(&MESSGE) +
0035.00
                        TOPGMO(*EXT) MSGTYPE(*STATUS)
0036.00
                       COND(&MSG#ID *EQ ,
0037.00
                        MSGQ(&LIBARY/&MSGQUE) MSGKEY(&MSGKEY))
                              CMDLBL(L1)
0039.00 PGMEND:
```

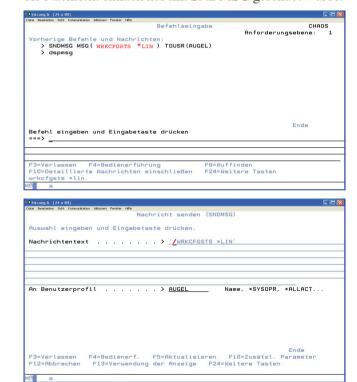
Man könnte weitere Funktionen hinzufügen, wie: "Ist Stelle 1 ein §, dann gib den Rest als MSG in einem Fenster aus." oder "Ist aus Stelle … der Wert …, dann mache …" und so weiter. Zentrale Steuer-Funktionen können so zentral und jederzeit ausgeführt werden ohne dass andere Programme auf die Fortsetzung warten müssen oder irgendwelche Asynchron-Jobs im Batch laufen, lauern oder warten.

Teil 3: Ausführung und Ergebnis

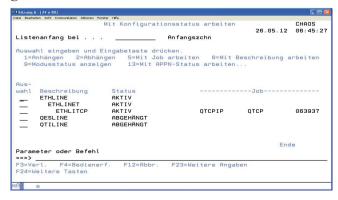
Ausführung: SNDMSG MSG('WRKCFGSTS *LIN')



Ergebnis: Die gesendete Nachricht stört nicht mehr den Ablauf durch Verdecken der Anzeige, sondern wird jetzt als Status-Nachricht in der 24. Zeile eingeblendet, obwohl die Nachricht einfach nur mit SNDMSG gesendet wurde.



Ergebnis: Msg wird als Befehl beim Empfänger ausgeführt.



25

NEWSolutions – Produktmonografie August/September 2012