

6

Einführung in die Nachrichtenverwaltung

Von der AS/400 wird oft behauptet, dass „ihr gesamtes Betriebssystem durch Nachrichten gesteuert wird“. Natürlich ist diese Aussage viel zu allgemein, um nicht in Frage gestellt zu werden. Trotzdem trifft sie weitgehend zu. Zumindest scheint es so, als ob Nachrichten das System dominieren. Auf Fehlerbedingungen wird mit Hilfe von Nachrichten hingewiesen. Job-Protokolle bestehen aus Nachrichten. Die Diagnose am Ende von Kompilierungslisten besteht aus Nachrichten. Die Befehle, die vom Systemabfragemenü aus gestartet werden, finden Sie in einer Nachricht. Sogar die Optionen von AS/400-Menüs sind Nachrichten.

Zumeist werden Nachrichten jedoch als Daten benutzt, die für die Kommunikation verwendet werden. Anders als in einer Datenbank gespeicherte Daten kann jede Nachricht in Bezug auf Format, Inhalt und Bedeutung einzigartig sein. Nachrichten dienen der Kommunikation mit den Benutzern, Programmierern, System-Operatoren, Programmen, Jobs und dem System-Historie-Protokoll.

Die OS/400-Nachrichtenbearbeitungsfunktion wird durch ein einziges Element, den Generalized Message Handler unterstützt. Dieser ist für die Verwaltung aller Nachrichtenverwaltungsfunktionen zuständig. Es stehen zwei Objekttypen, die Nachrichtendatei (MSGF) und die Message Queue (MSGQ) zur Verfügung, um den Message Handler zu unterstützen.

Nachrichtendateien

Der Inhalt einer Nachricht kann dynamisch als „Intermediate Message“ (die manchmal auch als Inprompt Message bezeichnet wird) zur Verfügung gestellt werden. Eine der Stärken der OS/400-Nachrichtenbearbeitung liegt darin, dass der Inhalt einer Nachricht partiell oder vollständig in einer Nachrichtenbeschreibung vordefiniert werden kann. Nachrichtenbeschreibungen werden in einer Nachrichtendatei gespeichert. OS/400 enthält eine Reihe von Nachrichtendateien, von denen auch jeweils eine für

- allgemeine Systemnachrichten (QCPFMSG) und für
- jedes Programmprodukt

verwendet wird. Alle Nachrichtendateien können mit Hilfe des Befehls Work with Message Files (WRKMSGF) angezeigt werden.

Der Name der Datei für allgemeine Systemnachrichten QCPFMSG stammt aus der Welt der S/38, der Vorgängerin der AS/400. QCPFMSG erhielt seinen Namen unter Berücksichtigung des Control Program Facility (CPF), des Betriebssystems der S/38. Aus Kompatibilitätsgründen wurde CPFMSG auf der AS/400 nicht umbenannt.

Zusätzlich zu den vom System zur Verfügung gestellten Nachrichtendateien wird für gewöhnlich mit Hilfe des Befehls Create Message File (CRTMSGF) mindestens eine Nachrichtendatei für benutzerdefinierte Nachrichten erstellt. Der folgende Befehl bewirkt, dass die Nachrichtendatei USERMSG in der Bibliothek für allgemeine Zwecke erstellt wird:

```
CRTMSGF  MASGF(QGPL/USERMSG)  TEXT('User-defined messages')
```

Benutzerdefinierte Nachrichtenbeschreibungen können USERMSG mit Hilfe der folgenden Befehle hinzugefügt werden. Sie dienen darüber hinaus auch der Verwaltung der Nachrichtenbeschreibungen.

Befehle:

- WRKMSGD (Work with Message Descriptions)
- ADDMSGD (Add Message Description)
- CHGMSGD (Change Message Description)
- RMVMSGD (Remove Message Description)
- DSPMSGD (Display Message Description)

Nachrichtenbeschreibungen

Da Benutzer und Programme unterschiedliche Anforderungen an Nachrichten stellen, enthalten vordefinierte Nachrichtenbeschreibungen oft sowohl Texte in natürlicher Sprache für User als auch Datenformate, die ein Programm problemlos verstehen kann. Zu der Definition einer Nachricht in einer Nachrichtendatei zählen:

- ein Message Identifier (MSGID), der dem Referenzieren der Nachricht dient
- der Nachrichtentext (First-Level-Text)
- zusätzlicher Text (Second-Level-Text)
- das Format für Substitutionsdaten, die in den Text eingefügt werden können
- weitere unterschiedliche Attribute.

Falls Sie „see spot run“ (ohne Anführungszeichen) im Bildschirm Command Entry eingeben würden und anschließend die Enter-Taste drücken würden, würde die folgende Nachricht erstellt werden:

```
> see spot run
      Command SEE in library *LIBL not found.
      Error found on SEE command.
```

Obwohl sie nicht aus vollständigen Sätzen bestehen, sind die Nachrichten doch informativ. Weiter bezieht sich jede Nachricht auf den Kontext, in dem sie erstellt wurde. Mit anderen Worten bezieht sich hier jede Nachricht auf den Befehl mit dem Namen „SEE“ (der nicht existiert). Wenn Sie den Cursor auf die zweite Nachricht stellen und die Hilfe-Taste drücken, werden zusätzli-

che Informationen über die Nachricht angezeigt. In den angezeigten Informationen wird auch der Message Identifier (CPF0001) sowie ein zusätzlicher Text für die Nachricht enthalten sein, der oft Ursachen von Fehlern und Vorschläge zu deren Behebung beinhalten kann.

Sie können die Beschreibung für den Message Identifier CPF0001 mit Hilfe des Befehls DSPMSG (Display Message Description) wie folgt anzeigen:

```
DSPMSG   CPF0001
```

Die Anzeige Display Formatted Message Text (Option 1 des DSPMSGD-Menüs) enthält den First-Level-Text und den Second-Level-Text dieser Nachricht. Auch die Positionen für zu ersetzende Textelemente können durch Nachrichtendatenfelder identifiziert werden. Nachrichtendatenfelder werden durch ein Ampersand (&) gekennzeichnet, auf das Zahlen von 1 bis 99 folgen. Nachrichtfelder erlauben es, dass Nachrichtendaten ersetzt werden können, um den letztendlichen Text der Nachricht spezifisch auf den Kontext auszurichten und somit informativer zu gestalten. Die Nachricht „Error found on SEE command“ sagt zum Beispiel weit mehr aus als „Error found on command“.

Die Nachricht CPF0001 enthält ein Nachrichtendatenfeld, das mit &1 gekennzeichnet ist. Die Position dieses Nachrichtenfeldes im Text der Nachricht bestimmt, wo der Name des Befehls platziert wird, wenn die Nachricht gesendet wird.

Die Anzeige Display Field Data (Option 2 des DSPMSG-Menüs) zeigt das Format aller Nachrichtendatenfelder für den Message Identifier an. Das einzige Nachrichtendatenfeld für CPF0001 ist &1. Es ist als Zeichenfeld (*CHAR) mit Länge 10 definiert. Es gibt keine sinnvolle Beschreibung für in der Nachrichtenbeschreibung enthaltene Nachrichtendatenfelder. Die Bedeutung des

Nachrichtendatenfeldes kann nur festgestellt werden, indem man beobachtet, wie es in der Nachricht angewandt wird.

Message Queues

Message Queues werden an ein Objekt gesendet bzw. von diesem empfangen, indem eine Message Queue (MSQ) benutzt wird. Es gibt verschiedene Arten von Message Queues. User Message Queues können mit Hilfe des Befehls Create Message Queue (CRTMSGQ) erstellt werden. Einige Queues wie zum Beispiel die System Operator Message Queue (QSYSOPR) und die System Log Message Queue (QHST) werden mit dem System geliefert.

Andere Message Queues wie Workstation Message Queues und Benutzerprofil-Message-Queues werden vom Betriebssystem dynamisch erstellt. Für jede Bildschirmeinheitenbeschreibung wird eine Workstation Message Queue erstellt. Für jedes Benutzerprofil des Systems wird ebenfalls eine Benutzerprofil-Message erstellt, es sei denn, es wird eine bereits existierende Workstation Message Queue oder eine User Message Queue als Benutzerprofil-Message-Queue zugewiesen.

Ein Job kann eine User Message Queue, eine Workstation Message Queue oder eine QSYSOPR zuordnen, wenn der Parameter Delivery Mode (DLVRY) des Befehls Change Message Queue (CHGMSGQ) benutzt wird. Übermittlungsmodi (Delivery Modes), die bewirken, dass eine Message Queue zugewiesen wird, sind *BREAK und *NOTIFY.

OS/400 erstellt automatisch eine besondere Message Queue, die als Job Message Queue bezeichnet wird. Ein großer Teil der Fehlerbearbeitung in CL-Programmen betrifft das Senden und Empfangen von Nachrichten in der Job Message Queue. Aus die-

sem Grund ist das Verstehen der Struktur der Job Message Queue als Voraussetzung für das Konzipieren von CL-Programmen anzusehen, die eine Fehlerbearbeitungslogik enthalten.

Sie können sich die Job Message Queue als interne Poststelle des Jobs vorstellen. Sie kann einen durchaus an eine Wand erinnern, die eine Unmenge von Briefkästen für die verschiedenen Abteilungen eines Unternehmens anbietet, um Memos zu verteilen, was im E-Mail-Zeitalter natürlich kein Thema mehr ist. Die Job Message Queue kann als Medium betrachtet werden, über das die Programme miteinander kommunizieren. In Abbildung 6.1 wird die Job Message Queue dargestellt. Weiter wird hier die Beziehung zwischen der Job Message Queue und dem Call Stack (Aufrufstapel) deutlich. Die Job Message Queue ist logisch in eine externe Message Queue für den Job und eine Call Message Queue für jeden Programm- oder Prozeduraufruf im Job unterteilt. Die externe Message Queue wird von Programmen benutzt, um mit dem User des Jobs zu kommunizieren.

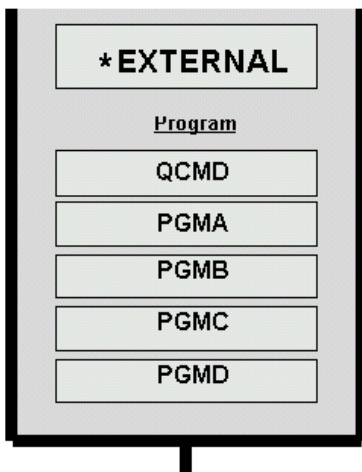


Abbildung 6.1: Job-Message-Queue-Stapel

Senden von Nachrichten

Für das Senden von Nachrichten an Message Queues stehen fünf Befehle zur Verfügung. In Tabelle 6.1 finden Sie die CL-Befehle, die bewirken, dass Nachrichten versandt werden. Darüber hinaus werden hier auch folgende Aspekte berücksichtigt:

- ob vordefinierte Nachrichten gesendet werden können
- die Arten von Message Queues, an die Nachrichten gesandt werden können
- die Nachrichtenarten, die gesendet werden können.

Tabelle 6.1: Befehle zum Senden von Nachrichten				
Befehlsname	Beschreibung	MSGID zulässig	MsgQ-Art	Art der Nachrichten
SNDMSG	Send Message	Nein	Non-PGM	*INFO *INQ
SNDBRKMSG	Send Break Message	Nein	WRKSTN	*INFO *INQ
SNDPGMMSG	Send Program Message	Ja	Alle	alle
SNDUSRMSG	Send User Message	ja	Non-PGM	*INFO *INQ
SNDRPY	Send Message Reply	nein	N/A	*RPY
Legende: Non-PGM - keine Call Message Queue				
WRKSTN - Workstation Device Message Queue				
N/A - Nicht anwendbar (not applicable)				

Die Befehle SNDMSG (Send Message) und SNDBRKMSG (Send Break Message) sind in Bezug auf das Senden von Nachrichten etwas beschränkt. Sie werden normalerweise von Benutzern und dem System-Operator benutzt, um Informationsnachrichten und Abfragenachrichten sofort an andere Benutzer weiterzugeben. Informationsnachrichten werden für gewöhnlich für die Kommunikation zwischen Benutzern verwendet. Abfragenachrichten werden benutzt, wenn eine Antwort gewünscht

wird. Eine Nachricht, für die der Befehl SNDBRKMSG benutzt wird, bewirkt, dass der interaktive Job der Workstation unterbrochen wird, wenn die Nachricht in den Job „einbricht“ und auf dem Bildschirm angezeigt wird. Abbildung 6.2 enthält einige Beispiele für die Befehle SNDMSG und SNDBRKMSG.

```
(1) SNDMSG      MSG( 'Please sign off NOW! ' ) TOUSR(*ALLACT)
(2) SNDMSG      MSG( '3 more minutes please! ' ) TOMSGQ(QSYSOPR)
(3) SNDMSG      MSG( 'I 'm making history. ' ) TOMSGQ(QHST)
(4) SNDBRKMSG   MSG( 'How about lunch? ' ) TOMSGQ(DSP09)  +
                MSGTYPE(*INQ)  RPYMSGQ(AARONV)
```

Abbildung 6.2: Beispiele für die Befehle SNDMSG und SNDBRKMSG

In Abbildung 6.2 sendet der erste SNDMSG-Befehl die Nachricht 'Please sign off NOW!' an alle User, die zu diesem Zeitpunkt auf dem System angemeldet sind. Die Nachrichten können, müssen aber die interaktiven Jobs nicht unterbrechen, was jedoch vom Übermittlungsmodus ihrer Message Queue abhängt. Der zweite Befehl bewirkt, dass die Informationsnachricht '3 more minutes please!' an die Message Queue des System-Operators gesendet wird. Diese Nachricht wird einen Job unterbrechen, wenn für die Message Queue QSYSOPR der Modus Break Delivery definiert ist.

Der dritte Befehl platziert die Eintragung 'I'm making history.' im System-Historie-Protokoll. Die Message kann angezeigt werden, wenn der Befehl DSPLOG (Display History Log) gestartet wird. Beachten Sie bitte den doppelten Apostroph in dem Wort 'I'm' (I'm) in Abbildung 6.2.

Der vierte SNDBRKMSG-Befehl sendet die Abfragenachricht 'How about lunch?' an die Workstation Message Queue DSP09.

Eine Kopie der Nachricht wird an AARONV, die Antwort-Message-Queue gesendet. Wenn auf der Workstation DSP09 ein Benutzer angemeldet ist, wird die Nachricht den interaktiven Job unterbrechen und der entsprechende User kann darauf antworten. Die Antwort wird in AARONV an die zugesandte Kopie der Nachricht angehängt. Falls die Message Queue AARONV von einem interaktiven Job in den Übermittlungsmodus *BREAK versetzt wurde, wird die Antwort diesen Job unterbrechen. Ist das nicht der Fall, kann die Antwort mit Hilfe des Befehls DSPMSG AARONV angezeigt werden.

Send User Message (SNDUSRMSG) ist in CL-Anwendungsprogrammen sehr nützlich. Wenn dieser Befehl benutzt wird, um eine Abfragenachricht zu senden, werden die Befehlsparameter zum Empfangen der Antwort in den Befehl SNDUSRMSG integriert. Darüber hinaus werden Parameter zur Angabe von Validierungskriterien für die Antwort eingefügt. Der folgende Befehl gibt dem System-Operator die Möglichkeit, die Verarbeitung fortzusetzen oder eine Task zu stornieren:

```
SNDUSRMSG  MSG('Go or Cancel?  (G C') VALUES(G C) +  
            TOMSGQ(QSYSOPR)  MSGRPY(&RPY)
```

Hier werden nur die Werte G und C als Antwort akzeptiert: Antworten in Kleinbuchstaben werden automatisch in Großbuchstaben übersetzt. Das Senden von CL-Programmen bleibt solange unterbrochen, bis der Operator auf die Nachricht antwortet. Sobald das geschehen ist, wird die CL-Variable &RPY die Antwort des Operators enthalten und die Programmlogik kann auf diesem Wert basieren.

Der vielseitigste Befehl für das Senden von Nachrichten ist SNDPGMMSG (Send Program Message). Er kann benutzt werden, um eine beliebige Art von Nachrichten an einen beliebigen

Message-Queue-Typ zu senden. Mit seiner Hilfe können entweder unmittelbare Nachrichten (Immediate Messages) oder vordefinierte Nachrichten versendet werden. SNDPGMMSG verfügt über eine Vielzahl von Parametern, die verschiedene Optionen für das Senden von Nachrichten bieten.

SNDPGMMSG-Parameter

Der Parameter MSG (Message) akzeptiert bis zu 3000 Zeichen Nachrichtentext für eine Eilmeldung. Seine Alternativen MSGID (Message Identifier) und MSGF (Message File) können benutzt werden, um eine vordefinierte Nachricht anzugeben. Alle Substitutionsdaten, die für eine vordefinierte Nachricht erforderlich sind, werden mit Hilfe des Parameters MSGDTA (Message Data) zur Verfügung gestellt. Falls zum Beispiel der Message Identifier CPF0001 (Error found on &1 command) in QCPFMMSG gesendet wird, könnte der Wert „SEE“ für Nachrichtendaten verwendet werden, was dazu führen würde, dass der endgültige Text der Nachricht „Error found on SEE command“ lauten würde.

Der Parameter MSGTYPE (Message Type) verfügt über den Default-Wert *INFO, erlaubt aber dass alle Nachrichtenarten gesendet werden können. Bei Nachrichten, die dem Abfragen bzw. Benachrichtigen dienen wird der Parameter RPYMSGQ (Reply Message Queue) benutzt, um die Message Queue zu identifizieren, an die die Antwort gesendet werden soll.

Die drei Parameter TOPGMQ (To Call Stack Entry Message Queue), TOMSGQ (To Non-Program Message Queue) und TOUSER (To User Profile) dienen dazu, die Nachricht an die gewünschte Message Queue weiterzugeben. Falls TOMSGQ oder TOUSER angegeben wird, wird die Nachricht an die angegebene Non-Program Message Queue gesendet.

Für TOPGMQ können die beiden Werte Relationship Entry (bzw. Offset Entry) und Call Stack Entry angegeben werden. Für den ersten dieser Werte kann *SAME (benutze die Eingabe im zweiten Wert) angegeben oder *PRV (benutze den Requester der Eingabe im zweiten Wert) angegeben werden. Für TOPGMQ kann der Wert *EXT angegeben werden. Dadurch wird darauf hingewiesen, dass die Nachricht an die externe Job Queue des Jobs gesendet werden soll. Die externe Message Queue wird von Programmen in einem Job zum Kommunizieren mit dem Workstation-Operator eines interaktiven Jobs benutzt. An die externe Message Queue des Jobs gesendete Nachrichten werden im Bildschirm Program Messages angezeigt.

In dem zweiten Wert (Call Stack Entry) kann der Name eines OPM-Programms, das sich derzeit im Call Stack (Aufrufstapel) des Jobs befindet, oder der vordefinierte Default-Wert „*“ benutzt werden. (* impliziert den Namen des OPM-Programms, das die Nachricht sendet). Auf die Anwendung der Aufrufstapelintragung und die Verwendung weiterer Sonderwerte in der ILE-Umgebung werde ich in Kapitel 13 eingehen. Die folgende Diskussion wird aus der OPM-Perspektive geführt.

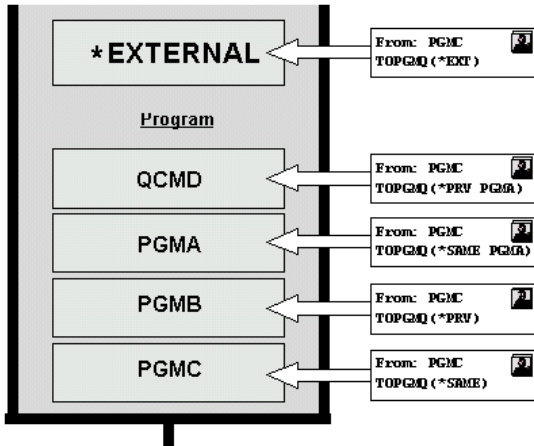


Abbildung 6.3: Das Ziel gesendeter Program Messages

Die zulässigen Werte für den Relationship-Teil von TOPGMQ weisen daraufhin, dass die Kommunikation zwischen Programmen nur in eine Richtung betrieben wird - nach oben. Da Programme immer unterbrochen werden, während andere Programme laufen, die sie aufgerufen haben, ist das aktive Programme, das die Steuerung inne hat, immer das unterste Programm im Aufrufstapel und kann deshalb keine Nachrichten an ein Programm senden, das im Aufrufstapel unter ihm platziert ist. Wenn der Parameter TOPGMQ des Befehls SNDPGMSG benutzt wird, kann ein CL-Programm Nachrichten senden an:

- sich selbst
- das Programm, von dem es aufgerufen wurde.
- jedes benannte Programm im Stapel
- jedes Programm, das ein benanntes Programm aufgerufen hat.
- die externe Message Queue.

Abbildung 6.3 zeigt, wie der Parameter TOPGMQ angegeben werden kann. In diesem Beispiel wird davon ausgegangen, dass der Aufrufstapel aus QCMD, PGMA, PGMB und PGMC besteht. PGMC ist das aktive Programm, das für die Steuerung zuständig ist. Es benutzt den Befehl SNDPGMMSG, um wie im folgenden Beispiel Nachrichten zu senden:

```
SNDPGMMSG MSG('ALL OK?') TOPGM(*EXT) MSGTYPE(*INQ)
```

Dieser Befehl bewirkt, dass die Abfragenachricht 'All OK?' an die externe Message Queue des Jobs gesendet wird. Der Job wird unterbrochen, während die Programmnachrichten angezeigt werden. Nachdem der Workstation-Operator auf die Nachricht geantwortet hat, wird die Steuerung an PGMC zurückgegeben. Nachrichten, die der Abfrage oder der Benachrichtigung dienen und an die externe Message Queue gesendet werden, geben dem Workstation Operator Gelegenheit, auf die Nachricht zu antworten.

Informationsnachricht, Ausführungsnachrichten und Diagnosenachrichten, die an die externe Message Queue gesendet wurden, werden solange angezeigt, bis der Operator die Enter-Taste drückt. Statusnachrichten, die an die externe Message Queue gesendet wurden, werden in der Nachrichtenzeile des aktuellen Bildschirm angezeigt, ohne den Job zu unterbrechen. Durchbruchnachrichten können nicht an den externen Aufrufer gesendet werden.

Abfragenachrichten können an die externe Message Queue gesendet werden, werden aber nicht angezeigt. Sie verbleiben als „schwebende Abfrage“ in der Job Message Queue, bis sie von einem Programm empfangen wird. Wenn ein Befehlseingabebildschirm angezeigt wird, werden schwebende Abfragen in Empfang genommen und wie ein zu startender Befehl behandelt. Abfragenachrichten können abgerufen werden, indem in einer

Befehlszeile oder in einem Befehlseingabebildschirm F9 gedrückt wird.

Der folgende Befehl bewirkt, dass eine Informationsnachricht an den Aufrufer von PGMA (QCMD) gesendet wird. Hier ist es erforderlich, den Namen des Elements zu kennen, das das Programm aufgerufen hat, um ihm eine Nachricht zu senden. Die Nachricht 'ALL OK' wird im Job-Protokoll erscheinen:

```
SNDPGMMSG MSG('ALL OK') TOPGMQ(*PRV PGMA) MSGTYPE(*INFO)
```

In diesem Beispiel kann der Parameter MSGTYPE weggelassen werden, da es sich bei *INFO um den Default-Wert handelt.

Der folgende Befehl sendet eine Durchbruchnachricht an PGMA, wodurch der Aufrufstapel zusammenbricht. Sowohl PGMC als auch PGMB werden beendet und PGMA wird über eine Ausnahmebedingung benachrichtigt:

```
SNDPGMMSG MSGID(CPF9897) MSGF(QCPFMSG) MSGDTA('Not OK') +  
TOPGM(*SAME PGMA) MSGTYPE(*ESCAPE)
```

Da eine vordefinierte Nachricht für eine Ausnahmebedingungs-
nachricht erforderlich ist, sind die Parameter MSGID und MSGF
in dem oben angeführten Befehl erforderlich. In diesem Beispiel
enthält der Message Identifier CPF0897 den Character String,
der im Parameter MSGDTA zur Verfügung gestellt wird, um die
Fehlernachrichten 'NOT OK!' zu senden. Viele Programmierer
werden eher mit der Nachricht CPF9898 vertraut sein. Der Un-
terschied zwischen beiden Nachrichten besteht darin, dass der
Nachrichtentext für CPF9898 einen Punkt nach der Substituti-
onsvariablen („&1.“) enthält. Falls im vorangegangenen Beispiel
CPF9898 angegeben worden wäre, würde 'NOT OK!' gesendet
werden.

Der folgende Befehl bewirkt, dass die Ausführungsnachricht 'ALL OK' an das Programm gesendet, das das sendende Programm aufgerufen hat (PGMB):

```
SNDPGMMMSG MSG('ALL OK') TOPGMQ(*PRV) MSGTYPE(*COMP)
```

Eine andere Möglichkeit, die To-Message-Queue als das Element anzugeben, das das sendende Programm aufgerufen hat, wäre TOPGMQ(*PRV*) anzugeben. Da es sich bei (*PRV*) um den Default-Wert handelt, könnte auf den Parameter TOPGMQ auch ganz verzichtet werden.

Falls der Wert der CL-Zeichenvariablen &ORD# gleich '001738' ist, bewirkt der folgende Befehl, dass die Informationsnachricht 'Ord#=001738' an die Call Message Queue des Senders gesendet wird:

```
SNDPGMMMSG MSG('Ord# =' *BCAT &ORD#) TOPGMQ(*SAME) MSGTYPE(*INFO)
```

Wenn Sie das Job-Protokoll durchsehen, werden Sie feststellen, dass die Order 001738 von dem Job verarbeitet wird.

Wenn der Befehl SNDPGMMMSG benutzt wird, um eine Nachricht zu senden, kann der MRK (Message Reference Key) an das sendende Programm zurückgegeben werden. Der MRK ist eine einmalige Zeichenkette, die ein bestimmtes Exemplar einer Nachricht in einer Queue identifiziert. Um den MRK zu empfangen, muss eine CL-Programmvariable im Parameter KEYVAR von SNDPGMMMSG definiert werden. Auf den MRK kann später im CL-Programm Bezug genommen werden, um die Nachricht zu empfangen, zu beantworten, zurück zu senden oder zu löschen. Nach dem folgenden SNDPGMMMSG-Befehl wird der Message Reference Key der Nachricht in der CL-Variablen &MRK gespeichert:

```
SNDPGMMMSG MSG('Save my key.') KEYVAR(&MRK)
```