

**Fujitsu Software BS2000 ASTI**  
Version V21.0B  
Mai 2022



## Readme-Datei

---

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

Copyright © 2024 Fujitsu

Die Marke Fujitsu und das Fujitsu Logo sind registrierte Marken der Fujitsu Limited, Japan in Europa und in anderen Ländern.

<b>1 Allgemeines</b>	<b>2</b>
<b>2 Betrieb</b>	<b>3</b>
<b>3 Kommandos</b>	<b>4</b>
3.1 Kommando-Returncodes	4
3.2 Service-Management-Kommandos	4
3.2.1 /START-SERVICE	4
3.2.2 /STOP-SERVICE	8
3.2.3 /SHOW-SERVICE-STATUS	8
3.2.4 /MODIFY-SERVICE-PARAMETER	8
3.3 Client-Kommandos	9
3.3.1 /SHOW-ORDER-STATUS	9
3.3.2 /DELETE-ORDER	10
3.3.3 /REQUEST-ORDER-RESULT	11
3.3.4 /SEND-ORDER	11
3.4 Service-Kommando	13
3.4.1 /PROCESS-ORDER	13
<b>4 Makros</b>	<b>15</b>
<b>5 Prozeduren</b>	<b>16</b>
5.1 Rücksetzen eines zentralen ASTI-Locks	16
5.2 Beispiel einer Service-Prozedur	17

# 1 Allgemeines

Das Subsystem ASTI ist eine Komponente, die einen Service verwaltet und die Kommunikation zwischen einer oder mehreren Client-Task(s) und einem Service (bestehend aus einer oder mehreren Tasks) bewerkstelligt.

ASTI ist ein Teil des Grundausbau und vorerst nur für interne Zwecke freigegeben. Da die Kommandos auf der Benutzeroberfläche erscheinen, werden sie in diesem Papier beschrieben.

\*4 Der Inhalt entspricht dem Freigabestand: Mai 2022.

\*1 Änderungen gegenüber Freigabestand Juli 2013 sind mit \*1 gekennzeichnet.

\*2 Änderungen gegenüber Freigabestand Juni 2014 sind mit \*2 gekennzeichnet.

\*3 Änderungen gegenüber Freigabestand März 2016 sind mit \*3 gekennzeichnet.

\*4 Änderungen gegenüber Freigabestand Januar 2021 sind mit \*4 gekennzeichnet.

Diese und andere aktuelle Readme-Dateien sind online verfügbar unter <https://bs2manuals.ts.fujitsu.com/index>.

## 2 Betrieb

ASTI ist kein eigenständiges System, das den Kunden direkt eine Leistung zur Verfügung stellt. Es ist gedacht für Subsystementwickler, die eine Art von Client-Server-Modell verwirklichen müssen. Hierbei stellt ASTI die Verbindung zwischen dem Client und dem Server her. Es können Verbindungen in alle Richtungen hergestellt werden (TU-TU, TPR-TU, TU-TPR und TPR-TPR). Der Client erteilt dem Server einen Auftrag, der vom Server abgearbeitet wird. Die Kommunikation zwischen den verschiedenen Tasks übernimmt ASTI.

ASTI nimmt den Auftrag und stellt ihn in eine Warteschlange für den Service. Es werden sämtliche Parameter gespeichert. Der Service fragt bei ASTI nach, ob ein Auftrag für ihn vorliegt. Ist einer vorhanden, so wird dieser an den Service weitergereicht. Der Service führt die Funktion aus und meldet das Ergebnis an den Client zurück.

Die Vorgänge können synchron, d.h. der Auftraggeber wartet auf das Ergebnis, oder asynchron abgegeben werden. Der Client kann auch steuern, wie lange die Aufträge im System aufgehoben werden müssen, bis eine Bearbeitung erfolgt. Ein Auftrag wird solange gespeichert, bis dieser Auftrag bearbeitet wurde. Wie lange das Ergebnis eines Auftrages gespeichert wird, kann ebenfalls der Client bestimmen.

Es gibt drei Möglichkeiten:

1. Der Auftraggeber möchte kein Ergebnis erhalten; ASTI verwirft das Resultat.
2. Der Auftraggeber möchte das Ergebnis während der Service-Session erhalten. Die Rückmeldung wird im Hauptspeicher gehalten und nach Service-Session-Ende gelöscht.
3. Der Auftraggeber möchte in jedem Fall ein Ergebnis erhalten. Die Rückmeldung wird auf der Platte gespeichert. Das Löschen erfolgt nach erfolgter Rückmeldung.

Das Subsystem ASTI verwaltet die Aufträge der Services in den Dateien \$TSOS.SYSQUE.ASTI und \$\*.SYS.<tsn>.<service-name>.<order-id>.

\*3 Das Format dieser Dateien hat sich mit ASTI V2.0B geändert. Solche Dateien mit Aufträgen aus ASTI V2.0A können in Nachfolgeversionen nicht mehr verarbeitet werden. Daher müssen diese Dateien vor einem Umstieg von ASTI V2.0A auf ASTI V2.0B oder eine Nachfolgeversion gelöscht werden. Das Subsystem ASTI darf nicht gestartet sein.

\$TSOS.SYSQUE.ASTI wird bei Start des Subsystems ASTI neu angelegt, wenn diese Datei nicht existiert.

Folgendes Vorgehen wird empfohlen:

```
/STOP-MAIL-SERVICE
```

```
/STOP-LWRES
```

\*1 /STOP-SUBSYSTEM REWAS

```
/STOP-SUBSYSTEM ASTI
```

```
/DELETE-FILE $TSOS.SYSQUE.ASTI
```

```
/DELETE-FILE $*.SYS.<tsn>.<service-name>.<order-id> (falls vorhanden)
```

## 3 Kommandos

### 3.1 Kommando-Returncodes

Alle Kommandos setzen Subcode 1, Subcode 2 und Maincode. Im Erfolgsfall bekommt der Maincode den Wert CMD0001. Im Fehlerfall enthält der Maincode den Schlüssel der Fehlermeldung, und es wird der Spin-Off-Mechanismus ausgelöst.

Die Service- und die Client-Kommandos setzen noch zusätzlich SDF-P-Variablen. Diese Variablen werden nur im Positivfall gesetzt:

```
SVTVAR-ORDERID
SVTVAR-SERVICE
SVTVAR-DATA
```

### 3.2 Service-Management-Kommandos

Diese Kommandos sind für die Steuerung des Services zuständig. Mit ihnen kann der Service gestartet und beendet werden. Außerdem kann der Status des Services ausgegeben und es können einige Modifikationen durchgeführt werden.

#### 3.2.1 /START-SERVICE

Dieses Kommando startet einen Service. Ein Service ist ein von ASTI erzeugter Batch-Job, der nur ein Programm startet. Das Programm wird durch ein Symbol und eine Library definiert. Wenn sich das Programm beendet, beendet sich auch der Job.

```
/START-SERVICE
```

```
SERVICE-NAME = <text_4..16>
```

```
,FROM-FILE = *LIBRARY-ELEMENT(..)
    (LIBRARY = <full-filename_1..54>
    ,SYMBOL = <composed-name_1..32_with_underscore>
        | <c-string_1..32_with_lower-case>)
    | *BY-IMON(..)
    (LOGICAL-ID = SYSLNK | <composed-name_1..30>
    ,INSTALLATION-UNIT = <composed-name_1..30>
    ,VERSION = *STD | <version 3..7>
    ,DEFAULT-LIBRARY = <full-filename_1..54>
    ,SYMBOL = <composed-name_1..32_with_underscore>)
    | *NONE(TSN = <text_4..4>)
    | *PROCEDURE(..)
    (<full-filename_1..54> | *LIBRARY-ELEMENT(..))
    (LIBRARY = <full-filename_1..54>
    ,ELEMENT = <composed-name_1..32_with_underscore>)
```

```
,PROCESSING-ADMISSION = *FROM-ORDER
    | *FROM-START
    | *EXPLICIT(..)
    (USERID = <name_1..8>
    ,ACCOUNT = <text_1..8>
    ,PASSWORD = *NONE | <secure1..8>)
```

```

,ALLOWED-USER = *ALL
    | *NONE
    | *START-USER-IDENTIFICATION(..)
*2      (ALTERNATE-ADMIN = *NONE | list_poss(8): <name_1..8>)
    | *BY-GUARD(NAME = <full-filename_1..24>)

,ORDER-RECOVERY = *PARAMETER(..)
    (ALLOWED = *NO | *SESSION-WIDE | *PERMANENT
    ,DEFAULT = *NO | *SESSION-WIDE | *PERMANENT)

,ORDER-LIMIT = 255 | <integer_1..32000>

,NUMBER-OF-TASKS = 1 | <integer_1..16>
    (MINIMAL = 0 | <integer_0..16>
    ,MAXIMAL = 16 | <integer_1..16>)

,START-PARAMETER = *NONE | <c-string_1..1800> | <x-string_1..3600>

,NOT-ALLOWED-CMD = *NONE |
    | list-poss(5): *STOP-SERVICE | *MODIFY-SERVICE-PARAM |
    *SEND-ORDER | *DELETE-ORDER |
    *REQUEST-ORDER-RESULT

,NOT-ALLOWED-SVC = *NONE |
    | list-poss(5): *SVTSTOP | *SVTMODI |
    *SVTSORD | *SVTDORD | *SVTRORD

,SYSTEM-OUTPUT = *ALL | *NONE | *STDOUT | *PARAMETER(..)
    ( SYSLST-OUTPUT = *STDOUT | *NONE | *PRINTER | *MAIL
    ,SYSOUT-OUTPUT = *STDOUT | *NONE | *PRINTER | *MAIL )

*2 ,MONJV = *NONE | <full-filename_1..24>
    
```

SERVICE-NAME	Ein für dieses System eindeutiger Name. Services, die mit \$ beginnen, können nur von TSOS gestartet werden.
FROM-FILE	Liefert die Ladebibliothek und das zu ladende Symbol. Es ist eine direkte Eingabe *LIBRARY-ELEMENT und eine indirekte Eingabe über den Installationsmonitor *BY-IMON möglich. *NONE ist eine Option, mit der eine laufende Dialog-Task als Service-Task angesehen wird. *PROCEDURE startet eine Prozedur als Service.
*LIBRARY-ELEMENT	
LIBRARY	Name einer Modulbibliothek
SYMBOL	Name einer CSECT oder eines ENTRY, mit dem ein Modul geladen und gestartet wird.
*BY-IMON	
LOGICAL-ID	Logischer Name der Modulbibliothek
INSTALL-UNIT	Name eines Produktes
VERSION	Version des Produktes
	*STD ist die höchste installierte Version.
DEFAULT-LIB	Default-Bibliotheksname, falls Produkt nicht installiert ist
SYMBOL	Name einer CSECT oder eines ENTRY, mit dem ein Modul geladen und gestartet wird.
*NONE	
TSN	Hier kann die TSN einer Dialog-Task eingegeben werden.

*PROCEDURE	
FILENAME	Name einer Prozedurdatei
LIBRARY	Name einer Modulbibliothek
ELEMENT	Name eines J-Elementes in der Bibliothek
PROCESSING-ADMISSION	regelt, in welcher User-Umgebung der Service arbeitet.
*FROM-ORDER	Der Service übernimmt die User-Umgebung der Task, welche die Order gesendet hat. Diese Funktion ist bei FROM-FILE=*NONE nicht erlaubt.
*FROM-START	Der Service erhält eine User-ID, die während des Betriebs nicht geändert wird.
*EXPLICIT	Angabe der User-ID, unter welcher der Service gestartet wird
USERID	Ablaufkennung des Services
ACCOUNT	Accountnummer der User-ID
PASSWORD	*NONE           Kein Passwort vergeben <secure_1..8>    Passwort der User-ID
ALLOWED-USER	regelt den Zugriff auf den Service
*START-USER-IDENTIFICATION(...)	Es wird nur die gleiche User-ID wie beim Start-Service zugelassen, oder eine der alternativen Kennungen aus der Liste.
*2           ALTERNATE-ADMIN	Eine Liste von bis zu acht (verschiedenen) Kennungen, die den Dienst ebenfalls verwalten können.
*BY GUARD	Die User-IDs werden über ein Guard zugelassen. Das Guard steht in der Start-User-ID. Länge des Guard (1-8 Zeichen)
*NONE	Der Service nutzt nur die Service-Verwaltung.
*ALL	Alle Tasks haben Zugang.
ORDER-RECOVERY=*PARAMETER(...)	Dieser Parameter beeinflusst den Wertebereich des Parameters ORDER-RECOVERY im Kommando /SEND-ORDER.
ALLOWED	Dieser Parameter bestimmt den erlaubten Wertebereich des Parameters ORDER-RECOVERY im Kommando /SEND-ORDER.
*NO	Die ORDER und das RESULT werden nur im Arbeitsspeicher gehalten. Beendet sich eine Client-Task oder der Service, so werden die entsprechenden Ordern negativ quittiert.
*SESSION-WIDE	Die ORDER wird solange im Speicher gehalten, bis der Auftrag abgearbeitet ist. Nicht bearbeitete Ordern und Resultate werden im Arbeitsspeicher gehalten und erst bei Session Ende (/SHUTDOWN oder /STOP-SUBSYSTEM ASTI) gelöscht.
*PERMANENT	Die ORDER und das RESULT werden auf Platte gespeichert und erst nach Abarbeitung der Order bzw. Abholung des Resultats gelöscht. Bei dieser Funktion können bei falscher Programmierung Auftragsleichen entstehen, die in Form von Dateien in denjenigen User-IDs bestehen bleiben, unter denen die Aufträge gesendet wurden.

DEFAULT	Legt den Standardwert für ORDER-RECOVERY fest.
ORDER-LIMIT	Hiermit wird die maximale Anzahl der in Bearbeitung befindlichen Ordnern festgelegt.
NUMBER-OF-TASKS	Anzahl (1-16) der Service-Tasks für diesen Service
MINIMAL	Die minimale Anzahl (0-16) von Service-Tasks, die über /MODIFY-SERVICE-PARAMETER eingestellt werden kann.
MAXIMAL	Die maximale Anzahl (1-16) von Service-Tasks, die über /MODIFY-SERVICE-PARAM eingestellt werden kann.
START-PARAMETER	Dieser Parameter muss vom Service nach dem Start der Service-Task abgefragt werden. Der Satz dient zur Initialisierung des Services.
NOT-ALLOWED-CMD	Mit diesem Parameter können Kommandos für einen Service verhindert werden.
*STOP-SERVICE	Dieser Service kann nicht mit diesem Kommando beendet werden.
*MODIFY-SERVICE-PARAM	Dieser Service kann nicht modifiziert werden.
*SEND-ORDER	An diesen Service kann kein Auftrag erteilt werden.
*DELETE-ORDER	Es können keine Aufträge gelöscht werden.
*REQUEST-ORDER-RESULT	Es können keine Resultate abgeholt werden.
NOT-ALLOWED-SVC	Mit diesem Parameter können Programmaufrufe für einen Service verhindert werden.
*SVTSTOP	Dieser Service kann vom Programm nicht beendet werden.
*SVTMODI	Dieser Service kann vom Programm nicht modifiziert werden.
*SVTSORD	Vom Programm kann kein Auftrag gestellt werden.
*SVTDORD	Vom Programm können keine Aufträge gelöscht werden.
*SVTRORD	Vom Programm können keine Resultate abgeholt werden.
Anmerkung: TPR-Schnittstelle unterliegt keiner dieser Einschränkungen.	
SYSTEM-OUTPUT	setzt die Parameter beim EXIT-JOB
*ALL	SYSLST und SYSOUT werden ausgedruckt.
*NONE	Der Ausdruck wird unterdrückt.
*STDOUT	Das Verhalten ist in der Klasse-2-Option „SSMOUT“ spezifiziert.
*PARAMETER	
SYSLST-OUTPUT	Spezifiziert SYSLST
SYSOUT-OUTPUT	Spezifiziert SYSOUT
*STDOUT	Siehe Klasse-2-Option „SSMOUT“
*NONE	Kein Ausdruck der jeweiligen Datei
*PRINTER	Ausdruck der jeweiligen Datei
*MAIL	Versenden der Datei als Mail
*2 MONJV	Name einer JV, um die Task zu überwachen
*NONE	MONJV wird hier nicht verwendet.

Anmerkung: Es kann nur eine Task mit einer MONJV überwacht werden.

### 3.2.2 /STOP-SERVICE

Dieses Kommando stoppt einen Service. Der Service wird in den Termination-Mode gesetzt. Die einzelnen Tasks müssen sich selbst beenden. Neue Ordern werden nicht mehr entgegengenommen.

Noch vorhandene Ordern werden entsprechend der Einstellungen behandelt, die bei ihrem Versenden über den Parameter ORDER-RECOVERY vorgenommen wurden.

```
/STOP-SERVICE SERVICE-NAME= <text_4...16>
```

SERVICE-NAME                    Name des Services

### 3.2.3 /SHOW-SERVICE-STATUS

Mit diesem Kommando wird der Status eines einzelnen Service oder aller Services angezeigt. Diese Information kann auch in OPS-Variablen ausgegeben werden.

```
/SHOW-SERVICE-STATUS SERVICE-NAME = *ALL | <text_4..16>
```

#### Ausgabe für SERVICE-NAME=\*ALL

SID	SERVICE-NAME	USERID	STATUS	TASK	ALLOWED-USER	PRO-ADM
@006	SERVICE3	ASTITEST	RUNNING	0001	ALL	FROM-START
@005	SERVICE2	TSOS	RUNNING	0001	START-USERID	FROM-ORDER
@004	SERVICE1	TSOS	RUNNING	0001	ALL	FROM-ORDER
@003	MAILCLNT	TSOS	RUNNING	0001	ALL	FROM-START
@002	LWRESO	TSOS	RUNNING	0001	START-USERID	FROM-START
@001	\$ASTIQUE	TSOS	RUNNING	0001	ALL	FROM-START

#### Ausgabe für SERVICE-NAME=MAILCLNT

```
SID SERVICE-NAME STATUS ADMINSTR ALLOWED-USER PROCESSING MODE
@003 MAILCLNT RUNNING TSOS ALL FROM-START SERVICE
ORDER-RECOVERY: ALLOWED = PERMANENT DEFAULT = PERMANENT
SERVICE-TASKS : 01 LIMITS (MIN=00,MAX=16)
SYSTEM-OUTPUT : SYSOUT = *NONE SYSLST = *NONE
SYMBOL : MSSRVC
LIBRARY : :SQN6:$TSOS.SYSLNK.MAIL.032.BACKEND
GUARD : *NONE
TASK: 9UKX ACT-ORDER: 000 TERMINATION: NONE START-PARAM: NONE
% SVTS000 Service MAILCLNT: Command executed
```

### 3.2.4 /MODIFY-SERVICE-PARAMETER

```
/MODIFY-SERVICE-PARAMETER
```

```
SERVICE-NAME = <text_4..16>
```

```
,ALLOWED-USER = *UNCHANGED
| *ALL
| *START-USER-IDENTIFICATION
| *BY-GUARD(NAME=<full-filename_1..24>)
```

```
,NUMBER-OF-TASKS = *UNCHANGED | *PAST-VALUE | <integer_0..16>
```

```
,ORDER-LIMIT = *UNCHANGED | <integer_1..32000>
```

```
,START-PARAMETER = *UNCHANGED | *NONE
| <c-string_1...1800> | <x-string_1..3600>
```

SERVICE-NAME	Name des Services
ALLOWED-USER	Regelt, wer den Service nutzen darf.
NUMBER-OF-TASKS	Anzahl der Tasks eines Service
0-16	Die Task-Anzahl kann auf 0 reduziert werden. Der Service ist hierbei noch vorhanden und nimmt Ordnern an, führt sie jedoch nicht aus.
*PAST-VALUE	Bei einem Dialog-Service wird dieser Parameter ignoriert. Übernimmt den Wert des vorherigen /MODIFY-SERVICE-PARAMETER
ORDER-LIMIT	Hiermit wird die maximale Anzahl gleichzeitig vorhandener Ordnern festgelegt.
START-PARAMETER	Mit diesem Parameter können die Initialisierungsparameter geändert werden. Das nächste Kommando /PROCESS-ORDER des Services erhält den Returncode „SVTS011 neue Startparameter“ (x'00040010'). Steht zum Zeitpunkt der Parametermodifikation der Service mit /PROCESS-ORDER in der Wait-Börse, so wird der RC x'00040010' erst nach Börsen-Timeout geliefert.

### 3.3 Client-Kommandos

Diese Kommandos verbinden den Client mit dem Service.

#### 3.3.1 /SHOW-ORDER-STATUS

Mit diesem Kommando werden Informationen über Aufträge ausgegeben. Dieses Kommando arbeitet asynchron zum laufenden Betrieb, deshalb ist die Konsistenz zwischen der Anzahl der Einträge und den einzelnen Subqueues nicht gewährleistet.

Diese Information kann auch in OPS-Variablen ausgegeben werden.

/SHOW-ORDER-STATUS

```
INFORMATION = *SUMMARY ( SERVICE-NAME = *ALL | <text_4...16> )
              *ORDER-LIST ( SERVICE-NAME = *ALL | <text_4...16> )
              *ALL ( ORDER-ID = <x-text_1...16 > )
```

INFORMATION

*SUMMARY	Anzahl der Aufträge je Subqueue
*ORDER-LIST	Alle relevanten Order-Ids inkl. Status
*ALL	Informationen über die Ordnern (Administrator: über alle, normaler User: nur über seine eigenen)

SERVICE-NAME Name des Services

ORDER-ID Auftragsidentifikation  
Während derselben ASTI-Session genügt die Angabe der letzten acht Bytes, bei Aufträgen aus älteren Sessions sind alle sechzehn Bytes erforderlich.

Ausgabe für INFORMATION=\*SUMMARY

SERVICE	ALL-Q	RDY-Q	ACT-Q	RES-Q	WAI-Q	NRR-Q	IAC-Q	DEQ-R
SERVICE4	3	1	1	1	0	0	0	0

ALL-Q Alle Ordern dieses Service  
 RDY-Q Alle Ordern, die auf Ausführung durch den Service warten  
 ACT-Q Alle Ordern, die gerade ausgeführt werden  
 RES-Q Alle Resultate, die noch nicht abgeholt wurden  
 WAI-Q Alle Ordern, die zurückgestellt wurden  
 NRR-Q Alle Resultate, die nicht angefordert wurden  
 IAC-Q Ordern von beendeten Services (session-wide)  
 DEQ-R Resultate, deren Tasks nicht mehr vorhanden sind

Ausgabe für INFORMATION=\*ORDER-LIST

ORDER-ID	SERVICE	QUEUE	TASK
1266CAAA00000007	SERVICE4	READY-QUEUE	0BA4
1266CAAA00000006	SERVICE4	ACTIVE-QUEUE	0BA4
1266CAAA00000005	SERVICE4	RESULT-QUEUE	0BA4

% SVTS000 Service SERVICE4: Command executed

Ausgabe für INFORMATION=\*ALL

ORDER	SERVICE	CURRENT QUEUE	
1266CAAA00000007	SERVICE4	READY-QUEUE	
USERID	ACCOUNT	TSN	USER-STATE
TSOS	ADMINSTR	0BA4	RESULT-REQUESTED

Erläuterung:

USERID User-ID des Senders  
 ACCOUNT Account-Nummer des Senders  
 TSN TSN der Sender-Task  
 USER-STATE Status Resultat Anforderung

**3.3.2 /DELETE-ORDER**

Mit diesem Kommando wird eine Order oder alle Ordern eines Service gelöscht. Ein normaler User kann nur die Aufträge seiner Kennung, der Service-Administrator alle Ordern dieses Service löschen.

/DELETE-ORDER

```
ORDER-ID = *TASK(SERVICE-NAME = <text_4..16>
            |*USER(SERVICE-NAME = <text_4..16>
            |*ALL(SERVICE-NAME = <text_4..16>
            |<x-text_1..16>
```

ORDER-ID Auftragsidentifikation  
 Während derselben ASTI-Session genügt die Angabe der letzten acht Bytes, bei Aufträgen aus älteren Sessions sind alle sechzehn Bytes erforderlich.  
 Es werden nur Ordern gelöscht, die von der gleichen Task stammen.  
 Ausnahme: Permanente Ordern, deren Aufrufer nicht mehr vorhanden ist.

*TASK	Alle Ordern dieser Task an den genannten Service werden gelöscht, auch die mit permanenter Recovery-Anforderung.
*USER	Alle Ordern von dieser User-ID an den genannten Service werden gelöscht, auch die mit permanenter Recovery-Anforderung.
*ALL	Alle Ordern der User-ID an den genannten Service werden gelöscht. Ist die User-ID Administrator des Services, werden alle Aufträge gelöscht.
SERVICE-NAME	Name des Services

**Anmerkung:**

Es werden alle betroffenen Ordern gesucht und mit "order to delete" gekennzeichnet. In Bearbeitung befindliche Ordern werden noch abgearbeitet und das Resultat dem Auftraggeber zugestellt. Ist kein Resultat verlangt, wird das Resultat gelöscht. Aufträge in der Ready-Queue oder Wait-Queue werden gelöscht. Wartet der Auftraggeber auf ein Resultat, wird von ASTI ein negatives Resultat zugestellt (SVTS903).

**3.3.3 /REQUEST-ORDER-RESULT**

Mit diesem Kommando wird ein Resultat eines asynchronen Auftrags abgeholt und das Resultat in der ASTI-Verwaltung gelöscht. Es können auch Ergebnisse von Aufträgen mit permanenter Recovery-Anforderung, deren Aufrufer-Task nicht mehr vorhanden ist, abgeholt werden.

/REQUEST-ORDER-RESULT

```
ORDER-ID= <x-text_1..16>
          | *TASK(SERVICE-NAME=<text_1..16>)
          | *USER(SERVICE-NAME=<text_1..16>)
*4 ,WAIT-FOR-RESULT = *YES (TIME-LIMIT= *NO | <integer_1..2147483647>)
          | *NO
```

ORDER-ID Auftragsidentifikation  
Während derselben ASTI-Session genügt die Angabe der letzten acht Bytes, bei Aufträgen aus älteren Sessions sind all sechzehn Bytes erforderlich.

\*TASK Es wird die nächste Order dieser Task angefordert.  
\*USER Es wird die nächste Order dieser User-ID angefordert.  
SERVICE-NAME Auswahl des Services

WAIT-FOR-RESULT Angabe, ob auf das Ergebnis gewartet werden soll.  
TIME-LIMIT begrenzt die Wartezeit.

\*4 Der Parameterwert \*USER für ORDER-ID erlaubt auch die Resultate von Aufträgen abzuholen, denen beim Senden (/SEND-ORDER) die Abholberechtigung \*USER zugeordnet wurde oder deren Sender-Task nicht mehr vorhanden ist.

**3.3.4 /SEND-ORDER**

Mit diesem Kommando wird ein Auftrag erteilt.

/SEND-ORDER

```
SERVICE-NAME = <text_4..16>
*4 ,WAIT-FOR-RESULT = *YES (TIME-LIMIT = *NO | <integer_1..2147483647>)
*4 | *NO (RESULT = *NO | *YES(...))
*4 REQUEST-PERMISSION = *STD | *USER
```

```
,ORDER-RECOVERY = *STD
                    | *NONE
                    | *SESSION-WIDE
                    | *PERMANENT
,DATA = *NO | <c-string 1...1800> | <x-string 1..3600>
```

SERVICE-NAME      Name des Services, an den der Auftrag gesendet werden soll

WAIT-FOR-RESULT

\*YES                Der Auftrag wird synchron ausgeführt, die Task wartet, bis der Auftrag beendet ist. Das Resultat wird angezeigt.

    TIME-LIMIT      Begrenzt die Wartezeit.

\*NO                 Der Auftrag wird angenommen und asynchron abgearbeitet.

    RESULT=\*NO      Es wird kein Resultat verlangt.

    RESULT=\*YES     Es wird ein Resultat verlangt. Das Ergebnis muss mit /REQUEST-ORDER-RESULT abgeholt werden.

\*4                    Die Berechtigung für das Abholen des Resultats ist beschränkt.

\*4                    REQUEST-PERMISSION

                      steuert die Abholberechtigung für das Resultat

\*STD                Es wird das Standardverfahren verwendet. Zunächst hat nur die sendende Client-Task die Berechtigung das Resultat abzuholen. Wenn die sendende Task terminiert wird und das Resultat aufgrund der Einstellung für ORDER-RECOVERY noch verfügbar ist, dann erhalten alle Client-Tasks, die unter derselben User-ID wie die terminierte Task laufen, die Berechtigung das Resultat abzuholen.

\*USER               Jede Client-Task unter derselben User-ID wie die sendende Task ist selbst dann berechtigt das Resultat abzuholen, wenn die sendende Task noch läuft.

ORDER-RECOVERY

\*STD                Es wird der bei /START-SERVICE angegebene Standardwert genommen.

\*NONE               Es werden keine besonderen Maßnahmen zur Transaktionssicherheit getroffen.

\*SESSION-WIDE     Während einer ASTI-Session wird die Order bis zur Quittung durch den Service, das Ergebnis bis zur Zustellung an den Auftraggeber aufgehoben.

                      Order und Ergebnis werden bei Beendigung von ASTI unabhängig vom Verarbeitungsstand gelöscht.

\*PERMANENT        Im Gegensatz zu \*SESSION-WIDE werden weder die Order noch das Resultat bei Beendigung des Services oder des Subsystems ASTI gelöscht.

                      Beim nächsten Start von ASTI können die Resultate abgeholt werden, die Aufträge werden bei Start des entsprechenden Service erneut zugestellt

DATA                Der Inhalt des Parameters stellt den Auftrag an die Service-Task dar.

### 3.4 Service-Kommando

#### 3.4.1 /PROCESS-ORDER

Mit diesem Kommando holt der Service Aufträge von ASTI ab und gibt Resultate zurück.

/PROCESS-ORDER

\*4

```

ACTION=  *GET-ORDER (WAIT-FOR-ORDER =
          *YES(TIME-LIMIT = *NO
                | < integer_1..2147483647>)
          | *NO
          *SEND-ACK (ORDER-ID=<x-text_1..16>
                    ,RETURN-DATA = *NONE
                        | <c-string_1..1800>
                        | <x-string_1..3600>)
          *SEND-NAK (ORDER-ID = <x-text_1..16>
                    ,RETURN-KEY = <text_7..7>)
                    ,RETURN-DATA = *NONE
                        | <c-string_1..1800>
                        | <x-string_1..3600>)
          *SEND-RETRY(ORDER-ID = <x-text_1..16>
                    ,WAIT-TIME = <integer_1..2000>
                    ,RETURN-DATA = *NONE
                        | <c-string_1..1800>
                        | <x-string_1..3600>)
          *GET-START-PARAMETER
    
```

ACTION=

*GET-ORDER	Anfordern eines neuen Auftrags
WAIT-FOR-ORDER	
*YES	Die Service-Task wartet, bis ein neuer Auftrag eintrifft. Der Parameter TIME-LIMIT begrenzt die Wartezeit.
*NO	Die Service-Task prüft, ob ein neuer Auftrag vorliegt, und arbeitet sofort weiter.
*SEND-ACK	Auftrag mit ORDER-ID wurde korrekt ausgeführt.
ORDER-ID	Auftragsidentifikation
RETURN-DATA	Rückgabeparameter
*NONE	Es wird kein Wert zurückgegeben.
*SEND-NAK	Auftrag mit ORDER-ID wurde mit Fehler abgebrochen.
ORDER-ID	Auftragsidentifikation
RETURN-KEY	spezifiziert die Art des Fehlers
RETURN-DATA	Rückgabeparameter
*NONE	Es wird kein Wert zurückgegeben.
*SEND-RETRY	Der Auftrag kann temporär nicht ausgeführt werden.
ORDER-ID	Auftragsidentifikation
WAIT-TIME	Zeit in Sekunden, nach der der Auftrag soll wieder gestartet werden soll. Die Wartezeit kann sich bis zum Börsen-Timeout (30 sec) verlängern.
RETURN-DATA	Rückgabeparameter
*NONE	Es wird kein Wert zurückgegeben.

**\*GET-START-PARAMETER**

Nach dem Service-Start oder nach Aufforderung mittels RC werden Initialisierungsparameter abgefragt.

## **4 Makros**

Für die Schnittstellen stehen auch Makros zur Verfügung.

## 5 Prozeduren

### 5.1 Rücksetzen eines zentralen ASTI-Locks

Wird eine Task, die einen Lock auf zentrale ASTI-Daten hält, abnormal beendet, kommt es zu einer Deadlock-Situation in ASTI, da der Lock nicht freigegeben wird. Bis zur Aufhebung dieses Locks können weder neue Aufträge angenommen noch vorhandene verarbeitet werden.

Wegen des fehlenden Anschlusses von ASTI an die Task-Terminierung kann dieses sehr seltene Problem nicht verhindert werden.

\*3 Um aber in einer solchen Situation einen Neustart des Systems zu vermeiden, wird zum Rücksetzen dieses Locks die Prozedur **SYSPRC.ASTI.210.RESET-LOCK** mit ausgeliefert.

Die Prozedur setzt das Privileg TSOS voraus.

Sie sollte immer aufgerufen werden, wenn eine Deadlock-Situation vermutet wird. Die Prozedur gibt eine entsprechende Meldung aus, führt aber selbst keine Modifikation durch.

\*3 Zum Ablauf der Prozedur SYSPRC.ASTI.210.RESET-LOCK:

- Lesen der zentralen ASTI-Tabelle mit der Lock-Information
- Abhängig von der Lock-Information wird eine entsprechende Meldung ausgegeben:
  1. keine Deadlock Situation: X'00000000' oder X'FFFFFFFF':  
Meldung: "ACTIVE-Q NOT LOCKED AT THE MOMENT"
  2. Bei sonstigen Inhalten werden folgende Meldungen ausgegeben:  
„ACTIVE-Q LOCKED BY TSN <tsn>“  
„RESET LOCK FOR EXISTING TSN <tsn>? (Y/N)“ oder  
„RESET LOCK FOR UNKNOWN TSN <tsn>? (Y/N)“
- Bei Eingabe von „Y“ wird das Programm HELGA aufgerufen, um die Sperre zurückzusetzen.

## 5.2 Beispiel einer Service-Prozedur

```
/SET-PROCEDURE-OPTION
/BEGIN-PARAMETER-DECLARATION
/END-PARAMETER-DECLARATION
/&*
/&* DO.SERVICE
/&* DIESE PROZEDUR VERWIRKLICHT EINEN DIENST
/&*
/ PROCESS-ORDER ACTION=*GET-ORDER(WAIT-FOR-ORDER=*YES)
/ IF-CMD-ERROR;END-IF
/ IF (MAINCODE() NE 'SVTS016')
/ IF-BLOCK-ERROR;END-IF
/ IMPORT-VARIABLE VARIABLE-NAME=SVTVAR-ORDERID
/ IMPORT-VARIABLE VARIABLE-NAME=SVTVAR-SERVICE
/ IMPORT-VARIABLE VARIABLE-NAME=SVTVAR-DATA
/LOOP: REPEAT
/ SHOW-VARIABLE VARIABLE-NAME=SVTVAR-SERVICE
/ IF (MAINCODE() EQ 'CMD0001')
/ PROCESS-ORDER ACTION=*SEND-ACK( -
/ ORDER-ID=&(SVTVAR-ORDERID), -
/ RET-DATA = 'OKAY')
/ IF-BLOCK-ERROR;END-IF
/ ELSE-IF (MAINCODE() EQ 'SVTS011')
/ PROCESS-ORDER ACTION=*GET-START-PARAMETER
/ IF-BLOCK-ERROR;END-IF
/ WRITE-TEXT TEXT='START-PAR: SVTVAR-DATA'
/ ELSE-IF (MAINCODE() NE 'SVTS016')
/ WRITE-TEXT TEXT='ERROR MAINCODE = &(MAINCODE())'
/ EXIT-BLOCK BLOCK=LOOP
/ END-IF
/ PROCESS-ORDER ACTION=*GET-ORDER(WAIT-FOR-ORDER=*YES)
/ IF-CMD-ERROR;END-IF
/ UNTIL (MAINCODE() EQ 'SVTS016')
/ END-IF
[/EXIT-PROC]
/&* muss ersetzt werden, um CMD2001 bei Task-Terminierung zu
/&* vermeiden.
/EXIT-JOB
```