

FUJITSU Software BS2000 ASTI

Version V2.0F
Juni 2016

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 © 2016 Fujitsu Technology Solutions GmbH

Die Marke Fujitsu und das Fujitsu Logo sind Marken oder registrierte Marken von Fujitsu Limited in Japan und in anderen Ländern. BS2000 ist eine Marke von Fujitsu Technology Solutions GmbH in Europa und in anderen Ländern.

1 Allgemeines	3
2 Betrieb	4
3 Kommandos	5
3.1 Kommando-Returncodes	5
3.2 Servicemanagement	5
3.2.1 Kommando: START-SERVICE	5
3.2.2 Kommando: STOP-SERVICE	9
3.2.3 Kommando: SHOW-SERVICE-STATUS	9
3.2.4 Kommando: MODIFY-SERVICE-PARAMETER	10
3.3 Client-Kommandos	11
3.3.1 Kommando: SHOW-ORDER-STATUS	11
3.3.2 Kommando: DELETE-ORDER	12
3.3.3 Kommando: REQUEST-ORDER-RESULT	13
3.3.4 Kommando: SEND-ORDER	13
3.4 Service-Kommando	14
3.4.1 Kommando: PROCESS-ORDER	14
4 Makros	16
5 Prozeduren	16
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 Grundaubaus und vorerst nur für interne Zwecke freigegeben. Da die Kommandos auf der Benutzeroberfläche erscheinen, werden sie in diesem Papier beschrieben.

*2 Der Inhalt entspricht dem Freigabestand: Juni 2016.

*2 Änderungen gegenüber Freigabestand Juni 2014 sind mit *1 gekennzeichnet.

*1 Änderungen gegenüber Freigabestand Juli 2013 sind mit *2 gekennzeichnet.

Diese und andere aktuelle Readme-Dateien sind auf der SoftBooks-DVD enthalten und online verfügbar unter <http://manuals.ts.fujitsu.com/>.

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 sowohl 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 3 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>. Das Format dieser Dateien hat sich geändert und kann ab ASTI V2.0B nicht mehr verarbeitet werden.

Vor einem Umstieg von ASTI V2.0A auf ASTI V2.0B müssen daher diese Dateien 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-LWRESO (ab BS2000/OSD-BC V9.0)

*1 STOP-SUBSYSTEM REWAS (ab BS2000 OSD/BC V10.0 auf SE Servern)

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 haben den SUBCODE1(), SUBCODE2() und den MAINCODE() gesetzt. Im Okay-Fall hat der MAINCODE den Wert CMD0001. Im Fehlerfall enthält der MAINCODE den Schlüssel der Fehlermeldung und es wird die SKIP-Funktion ausgelöst.

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

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

3.2 Servicemanagement

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

3.2.1 Kommando: START-SERVICE

Dieses Kommando startet einen Service. Ein Service ist ein von ASTI erzeugter Batchjob, der nur ein Programm startet. Das Programm wird durch ein Symbol und einer 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
,SYOUT-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 Dieser Parameter 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 ein laufender Dialogtask als Servicetask 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 eines 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 eine TSN eines Dialogtasks 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 des Tasks, der die Order gesendet hat. Diese Funktion ist bei FROM-FILE=*NONE nicht erlaubt.
*FROM-START	Der Service erhält eine Userid, die während des Betriebs nicht geändert wird.
*EXPLICIT	Angabe der Userid, unter der der Service gestartet wird
USERID	<text1..8> Ablaufkennung des Service
ACCOUNT	<text1..8> Accountnummer der Userid
PASSWORD	*NONE kein Password vergeben <secure1..8> Passwort der Userid
ALLOWED-USER	regelt den Zugriff auf den Service
*START-USER-IDENTIFICATION()	Es wird nur die gleiche Userid wie beim Start-Service zugelassen, oder eine der alternative Kennungen aus der Liste.
*2 ALTERNATE-ADMIN	Eine Liste von bis 8 (verschiedene) Kennungen die den Dienst ebenfalls verwalten können.
*BY-GUARD	Die Userids werden über ein Guard zugelassen. Das Guard steht in der Start-Userid. Länge des Guard (1..8 Char)
*NONE	Der Service nutzt nur die Service-Verwaltung
*ALL	Alle Tasks haben Zugang.
ORDER-RECOVERY	Dieser Parameter beeinflusst den Wertebereich des ORDER-RECOVERY-Parameters im SEND-ORDER Kommando.
*ALLOWED	Dieser Parameter bestimmt den erlaubten Wertebereich des ORDER-RECOVERY Parameters im SEND-ORDER Kommando
*NO	Die ORDER und das RESULT werden nur im Arbeitsspeicher gehalten. Beendet sich ein Client-Task oder der Service, so werden die entsprechenden Orders negativ quittiert. Resultate werden bei fehlender Order-Task gelöscht.
*SESSION-WIDE	Die ORDER wird solange im Speicher gehalten bis der Auftrag abgearbeitet ist. Nicht bearbeitete Order 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 der Order-Userid in Form von Dateien vorhanden bleiben.
DEFAULT	Dieser Parameter legt den Standard-Wert für ORDER-RECOVERY fest.
ORDER-LIMIT	Hiermit wird die maximale Anzahl der in Bearbeitung befindlichen ORDER festgelegt
NUMBER-OF-TASKS	1-16 Anzahl der Servicetask für diesen Service
MINIMAL	0-16 Die minimale Anzahl Servicetasks, die über /MODIFY-SERVICE-PARAM eingestellt werden können.
MAXIMAL	1-16 Die maximale Anzahl Servicetasks, die über /MODIFY-SERVICE-PARAM eingestellt werden können.
START-PARAM	Dieser Parameter muss vom Service nach dem Start des 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
* <u>ALL</u>	SYSLST und SYSOUT werden ausgedruckt
* <u>NONE</u>	Der Ausdruck wird unterdrückt
* <u>STDOUT</u>	Das Verhalten ist in der CLASS-II-Option „SSMOUT“ spezifiziert
*PARAMETER	
SYSLST-OUTPUT	spezifiziert SYSLST
SYSOUT-OUTPUT	spezifiziert SYSOUT
* <u>STDOUT</u>	siehe Class II Option „SSMOUT“
* <u>NONE</u>	Kein Ausdruck der jeweiligen Datei
* <u>PRINTER</u>	Ausdruck der jeweiligen Datei
* <u>MAIL</u>	Versenden der Datei als Mail

- *2 MONJV Der Name einer JV um die Task zu überwachen.
- *2 *NONE MONJV wird hier nicht verwendet.

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

*2

3.2.2 Kommando: STOP-SERVICE

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

Noch vorhandene Orders werden entsprechend dem beim Start angegebenen Wert für die ORDER-RECOVERY behandelt.

STOP-SERVICE SERVICE-NAME= <text_4...16> Name des Service

3.2.3 Kommando: 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 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 LWRES D TSOS RUNNING 0001 START-USERID FROM-START
@001 $ASTIQUE TSOS RUNNING 0001 ALL FROM-START
```

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

3.3 Client-Kommandos

Diese Kommandos verbinden den Client mit dem Service.

3.3.1 Kommando: SHOW-ORDER-STATUS

Mit diesem Kommando werden Informationen über Aufträge ausgegeben. Dieses Kommando arbeitet asynchron zum laufenden Betrieb, deshalb ist eine 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 incl. Status
*ALL	Informationen über die Order (Administrator für alle, normaler User nur auf seine eigenen Order)
SERVICE-NAME	Name des Service
ORDER-ID	Auftrags-Identifikation während derselben ASTI-Session genügt die Angabe der letzten 8 Bytes, bei Aufträgen aus älteren Sessions sind alle 16 Bytes erforderlich.

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

Ausgabe Information *ORDER-LIST

ORDER-ID	SERVICE	QUEUE	TASK
1266CAA000000007	SERVICE4	READY-QUEUE	0BA4
1266CAA000000006	SERVICE4	ACTIVE-QUEUE	0BA4
1266CAA000000005	SERVICE4	RESULT-QUEUE	0BA4

% SVTS000 Service SERVICE4: Command executed

Ausgabe Information *ALL

```
ORDER          SERVICE      CURRENT QUEUE
1266CAA0000007 SERVICE4    READY-QUEUE
USERID ACCOUNT  TSN          USER-STATE
TSOS  ADMINSTR 0BA4      RESULT-REQUESTED
```

Erläuterung:

```
USERID      Userid des Senders
ACCOUNT     Account des Senders
TSN         TSN der Sendertask
USER-STATE  Status Resultat Anforderung
```

3.3.2 Kommando: DELETE-ORDER

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

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 Auftrags-Identifikation
 während derselben ASTI-Session genügt die
 Angabe der letzten 8 Bytes, bei Aufträgen aus
 älteren Sessions sind alle 16 Bytes erforderlich.
 Es werden nur Order gelöscht, die vom gleichen Task stammen.

Ausnahme: Permanente Order, deren Aufrufer nicht mehr
 vorhanden ist.

```
*TASK(SERVICE-NAME=)      alle Order dieser Task an den genannten
                           Service werden gelöscht, auch die mit
                           permanenter Recovery Anforderung
*USER(SERVICE-NAME=)      alle Order von dieser Userid an den genannten
                           Service werden gelöscht, auch die mit
                           permanenter Recovery Anforderung
*ALL(SERVICE-NAME=)      alle Order der Userid an den genannten Service
                           werden gelöscht. Ist die Userid Administrator
                           des Services, werden alle Aufträge gelöscht.
```

Anmerkung:

Es werden alle betroffenen Order gesucht und mit "order to delete" gekennzeichnet. In Bearbeitung befindliche Order 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 Kommando: 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 Aufrufertask nicht mehr vorhanden, abgeholt werden.

REQUEST-ORDER-RESULT

```
ORDER-ID= <x-text_1..16>
          | *TASK(SERVICE-NAME=<text_1..16>)
          | *USER(SERVICE-NAME=<text_1..16>)
          ,WAIT-FOR-RESULT = *YES (TIME-LIMIT= *NO
                              | <integer_15..65535>)
                              | *NO
```

ORDER-ID	Auftrags-Identifikation während derselben ASTI-Session genügt die Angabe der letzten 8 Bytes, bei Aufträgen aus älteren Sessions sind alle 16 Bytes erforderlich.
*TASK	Es wird die nächste Order dieser Task angefordert
*USER	Es wird die nächste Order dieser Userid angefordert
SERVICE-NAME	Auswahl des Services
WAIT-FOR-RESULT	Angabe, ob auf das Ergebnis gewartet werden soll. TIME-LIMIT begrenzt die Wartezeit.

Mit dem Parameter *USER können auch Resultate, deren Task nicht mehr vorhanden ist, abgeholt werden.

3.3.4 Kommando: SEND-ORDER

Mit diesem Kommando wird ein Auftrag erteilt.

SEND-ORDER

```
SERVICE-NAME = <text_4..16>
              ,WAIT-FOR-RESULT = *YES (TIME-LIMIT = *NO
                                      | <integer_15..65535>)
                                      | *NO (RESULT = *NO | *YES )
              ,ORDER-RECOVERY = *STD
                                      | *NONE
                                      | *SESSION-WIDE
                                      | *PERMANENT
              ,DATA = *NO | <c-string 1...1800> | <x-string 1..3600>
```

SERVICE-NAME	Name der Service
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.

ORDER-RECOVERY	
*STD	Es wird der bei /START-SERVICE angegebene Defaultwert genommen.
*NONE	Es werden keine besonderen Maßnahmen zur Transaktions-Sicherheit 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 gelöscht, unabhängig vom Verarbeitungsstand.
*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 Services erneut zugestellt.
DATA	Der Inhalt des Parameters stellt den Auftrag an die Servicetask dar.

3.4 Service-Kommando

3.4.1 Kommando: PROCESS-ORDER

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

PROCESS-ORDER

```

ACTION=  *GET-ORDER (WAIT-FOR-ORDER =
                *YES(TIME-LIMIT = *NO | <integer_15..65535>)
                | *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
    
```

*GET-ORDER	Anfordern eines neuen Auftrags
WAIT-FOR-ORDER	
*YES	Die Servicetask wartet bis ein neuer Auftrag eintrifft. Der Parameter TIME-LIMIT begrenzt die Wartezeit
*NO	Die Servicetask prüft, ob ein neuer Auftrag vorliegt, und arbeitet sofort weiter.

- *SEND-ACK
 RETURN-DATA
 *NONE

Auftrag mit ORDER-ID wurde korrekt ausgeführt.
 Rückgabeparameter.
 Es wird kein Wert zurückgegeben.
- *SEND-NAK
 RETURN-KEY
 RETURN-DATA
 *NONE

Auftrag mit ORDER-ID wurde mit Fehler abgebrochen.
 spezifiziert die Art des Fehler
 Rückgabeparameter
 Es wird kein Wert zurückgegeben
- *SEND-RETRY
 WAIT-TIME

Der Auftrag kann temporär nicht ausgeführt werden.
 Zeit in Sekunden, nach der der Auftrag soll wieder
 gestartet werden soll. Die Wartezeit kann sich bis zum
 Boursen-Timeout (30sec) verlängern.

RETURN-DATA
 *NONE

Rückgabeparameter
 Es wird kein Wert zurückgegeben
- *GET-START-PARAMETER

Nach dem Servicestart oder nach Aufforderung
 mittels RC werden Initialisierungsparameter abgefragt.

4 Makros

Es stehen für die Schnittstellen 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 Taskterminierung kann dieses sehr seltene Problem nicht verhindert werden.

Um aber in einer solchen Situation einen Neustart des Systems zu vermeiden, wird mit ASTI ab V2.0C die Prozedur **SYSPRC.ASTI.020.RESET-LOCK** zum Rücksetzen dieses Locks mit ausgeliefert.

Die Prozedur setzt das TSOS Privileg 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.

Zum Ablauf der Prozedur SYSPRC.ASTI.020.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

```