HITACHI
Inspire the Next

For Windows Systems

OpenTP1 Version 7

User's Guide

3000-3-D64(E)

■ **Relevant program products**

For Windows Server 2003, Windows Server 2003 x64 Editions, Windows Server 2003 R2, Windows Server 2003 R2 x64 Editions, Windows XP, Windows Vista, Windows Vista x64 Edition, Windows Server 2008, Windows Server 2008 x64 Editions, Windows Server 2008 R2, and Windows 7

P-2464-2274 uCosminexus TP1/Server Base 07-03

P-2464-2374 uCosminexus TP1/FS/Direct Access 07-03

P-2464-2474 uCosminexus TP1/FS/Table Access 07-03

P-2464-2544 uCosminexus TP1/Extension 1 07-00

P-2464-3154 uCosminexus TP1/Message Control 07-03

P-2464-3254 uCosminexus TP1/NET/Library 07-04

P-2464-C374 uCosminexus TP1/Message Queue 07-01

P-F2464-3254C uCosminexus TP1/NET/TCP/IP 07-02

R-15452-21 uCosminexus TP1/Extension for .NET Framework 07-00

For Windows Server 2003, Windows Server 2003 x64 Editions, Windows Server 2003 R2, Windows Server 2003 R2 x64 Editions, Windows XP, Windows Server 2008, Windows Server 2008 x64 Editions, and Windows Server 2008 R2

P-F2464-32545 uCosminexus TP1/NET/XMAP3 07-01

For Windows Server 2003, Windows Server 2003 x64 Editions, Windows Server 2003 R2, Windows Server 2003 R2 x64 Editions, Windows Server 2008, Windows Server 2008 x64 Editions, and Windows Server 2008 R2

P-2464-2934 uCosminexus TP1/High Availability 07-00

P-F2464-3254D uCosminexus TP1/NET/High Availability 07-01

This manual can be used for products other than the products shown above. For details, see the *Release Notes*.

These products were developed under a quality management system that has received ISO9001 and TickIT certification.

■ **Trademarks**

■ Restrictions

Printed in Japan.

■ Edition history

Edition 1 (3000-3-D64(E)): July 2012

■ Copyright

Preface

This manual describes the differences in functions between the Windows editions of the uCosminexus OpenTP1 products listed below (referred to hereafter as *OpenTP1 for Windows*) and the UNIX editions of these uCosminexus OpenTP1 products (referred to hereafter as *OpenTP1 for UNIX*):

- P-2464-2274 uCosminexus TP1/Server Base
- P-2464-2374 uCosminexus TP1/FS/Direct Access
- P-2464-2474 uCosminexus TP1/FS/Table Access
- P-2464-2544 uCosminexus TP1/Extension 1
- P-2464-2934 uCosminexus TP1/High Availability
- P-2464-3154 uCosminexus TP1/Message Control
- P-2464-3254 uCosminexus TP1/NET/Library
- P-2464-C374 uCosminexus TP1/Message Queue
- P-F2464-32545 uCosminexus TP1/NET/XMAP3
- P-F2464-3254C uCosminexus TP1/NET/TCP/IP
- P-F2464-3254D uCosminexus TP1/NET/High Availability
- R-15452-21 uCosminexus TP1/Extension for .NET Framework

TP1/Message Queue was implemented on a foundation provided by the MQI, MQFAP, and MQ cluster specifications of WebSphere MQ, based on a licensing agreement with International Business Machines Corporation (IBM) of the U.S.

Intended readers

This manual is intended for system administrators, system designers, programmers, and operators of OpenTP1 for Windows.

Readers of this manual must have a basic knowledge of Windows and of online systems. Descriptions in this manual also assume that readers are familiar with the OpenTP1 series manuals listed below. We recommend that you use this manual in conjunction with reading the following manuals, as necessary:

- *OpenTP1 Version 7 Description* (3000-3-D50(E))
- *OpenTP1 Version 7 Programming Guide* (3000-3-D51(E))
- *OpenTP1 Version 7 System Definition* (3000-3-D52(E))

- *OpenTP1 Version 7 Operation* (3000-3-D53(E))
- *OpenTP1 Version 7 Programming Reference C Language* (3000-3-D54(E))
- *OpenTP1 Version 7 Programming Reference COBOL Language* (3000-3-D55(E))
- *OpenTP1 Version 7 Messages* (3000-3-D56(E))

Organization of this manual

This manual is organized into the following chapters and appendix:

1. Positioning of This Manual

Chapter 1 describes the relationship of this manual to the other manuals in the OpenTP1 series. It also recommends how you should use this manual.

2. Description of Functions

Chapter 2 describes the differences in program-level functions between OpenTP1 for Windows and OpenTP1 for UNIX, based on a listing of these functions. This chapter also provides detailed explanations of the functions that are specific to OpenTP1 for Windows.

3. Programming

Chapter 3 describes the differences in code-level functions between OpenTP1 for Windows and OpenTP1 for UNIX, based on a listing of these functions. This chapter also explains how to create programs that are specific to OpenTP1 for Windows.

4. Environment Setup

Chapter 4 explains the environment settings specific to OpenTP1 for Windows.

5. System Definitions

Chapter 5 describes the differences in the system definitions between OpenTP1 for Windows and OpenTP1 for UNIX, based on listings of system definitions. This chapter also provides detailed explanations of selected system definitions.

6. Operations

Chapter 6 explains the operation methods that are specific to OpenTP1 for Windows, as well as the graphical user interface (GUI) used by OpenTP1 for Windows.

7. Operation Commands

Chapter 7 describes the differences in the operation commands between OpenTP1 for Windows and OpenTP1 for UNIX, based on a listing of the operation commands. This chapter also provides detailed explanations of selected operation

commands.

A. Lists of OpenTP1 Processes

Appendix A provides lists of the system service processes used by OpenTP1 for Windows.

Related publications

This manual is part of a related set of manuals. The manuals in the set are listed below (with the manual numbers):

OpenTP1 products

- *OpenTP1 Version 7 Description* (3000-3-D50(E))
- *OpenTP1 Version 7 Programming Guide* (3000-3-D51(E))
- *OpenTP1 Version 7 System Definition* (3000-3-D52(E))
- *OpenTP1 Version 7 Operation* (3000-3-D53(E))
- *OpenTP1 Version 7 Programming Reference C Language* (3000-3-D54(E))
- *OpenTP1 Version 7 Programming Reference COBOL Language* (3000-3-D55(E))
- *OpenTP1 Version 7 Messages* (3000-3-D56(E))
- *OpenTP1 Version 7 Tester and UAP Trace User's Guide* (3000-3-D57(E))
- *OpenTP1 Version 7 TP1/Client User's Guide TP1/Client/W, TP1/Client/P* (3000-3-D58(E))
- *OpenTP1 Version 7 TP1/Client User's Guide TP1/Client/J* (3000-3-D59(E))
- *OpenTP1 Version 7 TP1/LiNK User's Guide* (3000-3-D60(E))#
- *OpenTP1 Version 7 Protocol TP1/NET/TCP/IP* (3000-3-D70(E))
- *OpenTP1 Version 7 TP1/Message Queue User's Guide* (3000-3-D90(E))#
- *OpenTP1 Version 7 TP1/Message Queue Messages* (3000-3-D91(E))#
- *OpenTP1 Version 7 TP1/Message Queue Application Programming Guide* (3000-3-D92(E))#
- *OpenTP1 Version 7 TP1/Message Queue Application Programming Reference* (3000-3-D93(E))#

Other OpenTP1 products

- *TP1/Web User's Guide and Reference* (3000-3-D62(E))#

Other related products

- *Indexed Sequential Access Method ISAM* (3000-3-046(E))
- *XP/W* (3000-3-047(E))
- *Extended Mapping Service 2/Workstation XMAP2/W DESCRIPTION/USER'S GUIDE* (3000-7-421(E))
- *SEWB 3 General Information* (3000-7-450(E))
- *Job Management Partner 1/Base User's Guide* (3020-3-K06(E))
- *Job Management Partner 1/Base Messages* (3020-3-K07(E))
- *Job Management Partner 1/Base Software Developer's Guide* (3020-3-K08(E))

For OpenTP1 protocol manuals, please check whether English versions are available.

#

If you want to use this manual, confirm that it has been published. (Some of these manuals might not have been published yet.)

This manual abbreviates manual titles. The following table lists the full name of each manual together with its abbreviated name as used in this manual:

Full name	Name used in this manual
OpenTP1 Programming Reference C Language	*OpenTP1 Programming Reference*
OpenTP1 Programming Reference COBOL Language	
Scalable Database Server HiRDB Version 7 Installation and Design Guide, for Windows(R) systems	*Scalable Database Server HiRDB Installation and Design Guide*
Scalable Database Server HiRDB Version 8 Installation and Design Guide, for Windows(R) systems	
Scalable Database Server HiRDB Version 7 UAP Development Guide	*Scalable Database Server HiRDB UAP Development Guide*
Scalable Database Server HiRDB Version 7 UAP Development Guide, for Windows(R) systems	
Scalable Database Server HiRDB Version 8 UAP Development Guide	

Conventions: Fonts and symbols

(1) Conventions: Attribute notation symbols

Attribute notation symbols are used to indicate conventions such as the scope of user-specified values:

Attribute notation symbol	Convention
~	A swung dash indicates that a user-specified attribute value must follow.
<< >>	Double paired angle brackets indicate the default for a user-specified value.
< >	Paired angle brackets indicate the syntax element for a user-specified value.
(( ))	Double paired parentheses indicate the scope of a user-specified value.

(2) Conventions: Fonts

The following table explains the fonts used in this manual:

Font	Convention
Bold	**Bold** type indicates text on a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example: • From the **File** menu, choose **Open**. • Click the **Cancel** button. • In the **Enter name** entry box, type your name.
Italics	*Italics* are used to indicate a placeholder for some actual text to be provided by the user or system. For example: • Write the command as follows: `copy` *source-file target-file* • The following message appears: `A file was not found. (file =` *file-name*`)` *Italics* are also used for emphasis. For example: • Do *not* delete the configuration file.
`Code font`	A `code font` indicates text that the user enters without change, or text (such as messages) output by the system. For example: • At the prompt, enter `dir`. • Use the `send` command to send mail. • The following message is displayed: `The password is incorrect.`

(3) Conventions: Syntax description symbols

The following table explains the syntax description symbols used in this manual:

Symbol	Convention
|	In syntax explanations, a vertical bar separates multiple items, and has the meaning of OR. For example: `abcdefg -h iii|jjj` means that, following the `-h` option, you can specify either `iii` or `jjj`.

Symbol	Convention
{ }	In syntax explanations, curly brackets indicate that only one of the enclosed items is to be selected. For example: `abcdefg {-h|-i}` means that you can specify either the `-h` option or the `-i` option.
[ ]	In syntax explanations, square brackets indicate that the enclosed item or items are optional. For example: `abcdefg [-h|-i]` means that you can specify `abcdefg`, `abcdefg -h`, or `abcdefg -i`.
. . .	In syntax explanations, an ellipsis (. . .) indicates that the immediately preceding item can be repeated as many times as necessary. For example: `abcdefg -h` *user-server-name* `[,` *user-server-name*`] . . .` means that you can specify as many user server names in the `-h` option as necessary.
__	In syntax explanations, underlined text indicates the value that the system assumes when the relevant operand or command parameter is omitted. For example: `abcdefg [-h {iii|jjj}]` means that, if omitted, `iii` is assumed to be the value specified for the `-h` option.
Δ	In syntax explanations, an upper-case delta character indicates a space. For example: *logical-file-name* Δ *physical-file-name* means that a space separates the logical file name and the physical file name.

(4) Conventions: Syntax elements

The following characters are used for explaining syntax elements in user-specified values:

Syntax element	Characters
Alphabetic characters	Alphabetic characters (`A` to `Z`, `a` to `z`) and the underscore (_)
Alphabetic characters and symbols	Alphabetic characters (`A` to `Z`, `a` to `z`), hash mark (`#`), at sign (`@`), and backslash (\)
Alphanumeric characters	Alphabetic characters and numeric characters (`A` to `Z`, `a` to `z`, `0` to `9`)
Alphabetic characters, symbols, and numeric characters	Alphabetic characters, symbols, and numeric characters
Special characters	Asterisk (`*`)
Unsigned integer	Numeric characters
Hexadecimal number	Numeric characters, `A` to `F`, and `a` to `f` (However, upper-case characters `A` to `F` are not permitted when numeric characters and lower-case characters `a` to `f` are expressly permitted.)
Identifier	An alphanumeric character string that begins with an alphabetic character

Syntax element	Characters
Symbolic name	A string consisting of a combination of alphanumeric characters and symbols that begins with an alphabetic character or symbol
Character string	An array of alphabetic characters
Path name	A string consisting of symbolic names, forward slashes (/), and periods (.) that conforms to the path name conventions of the operating system being used.
OpenTP1 file name	A character string of 14 or fewer characters consisting of a combination of alphabetic characters, numeric characters, periods (.), underscores (_), and at marks (@).

Conventions: Other conventions used in this manual

(1) Conventions: Abbreviations for product names

This manual uses the following abbreviations for product names:

Full name or meaning	Abbreviation		
COBOL2002 Net Client Suite	COBOL2002		
COBOL2002 Net Client Runtime			
COBOL2002 Net Developer			
COBOL2002 Net Server Suite			
COBOL2002 Net Server Runtime			
HiRDB/Parallel Server Version 7	HiRDB		
HiRDB/Single Server Version 7			
HiRDB/Parallel Server Version 8			
HiRDB/Single Server Version 8			
Java™	Java		
Job Management Partner 1/Automatic Job Management System 2 - Agent	JP1/AJS2 - Agent	JP1/AJS2	JP1
Job Management Partner 1/Automatic Job Management System 2 - Manager	JP1/AJS2 - Manager		
Job Management Partner 1/Automatic Job Management System 2 - View	JP1/AJS2 - View		

Full name or meaning	Abbreviation	
Job Management Partner 1/Automatic Job Management System 2 - Scenario Operation Manager	JP1/AJS2 - Scenario Operation Manager	JP1/AJS2 - Scenario Operation
Job Management Partner 1/Automatic Job Management System 2 - Scenario Operation View	JP1/AJS2 - Scenario Operation View	
Job Management Partner 1/Base		
Job Management Partner 1/Cm2/Operations Assist Agent	JP1/Cm2	
Job Management Partner 1/NETM/Audit - Manager	JP1/NETM/Audit	
Job Management Partner 1/NETM/DM Client	JP1/NETM/DM	
Job Management Partner 1/NETM/DM Manager		
Job Management Partner 1/NETM/DM SubManager		
Microsoft(R) MS-DOS(R)	MS-DOS	
Micro Focus Net Express(R)	Net Express	
Oracle Database 10g	Oracle	
Oracle Database 11g		
Oracle9i		
Microsoft(R) SQL Server 2005	SQL Server	
Microsoft(R) SQL Server 2008		
uCosminexus TP1/Client/J	TP1/Client/J	TP1/Client
uCosminexus TP1/Client/P	TP1/Client/P	
uCosminexus TP1/Client/W	TP1/Client/W	
uCosminexus TP1/Client/W(64)		
uCosminexus TP1/Client for .NET Framework	TP1/Client for .NET Framework	
uCosminexus TP1/Connector for .NET Framework	TP1/Connector for .NET Framework	
uCosminexus TP1/Server Base Enterprise Option	TP1/EE	
uCosminexus TP1/Server Base Enterprise Option(64)		
uCosminexus TP1/Extension 1	TP1/Extension 1	

Full name or meaning	Abbreviation
uCosminexus TP1/Extension 1(64)	
uCosminexus TP1/FS/Direct Access	TP1/FS/Direct Access
uCosminexus TP1/FS/Direct Access(64)	
uCosminexus TP1/FS/Table Access	TP1/FS/Table Access
uCosminexus TP1/FS/Table Access(64)	
uCosminexus TP1/High Availability	TP1/High Availability
uCosminexus TP1/High Availability(64)	
uCosminexus TP1/LiNK	TP1/LiNK
uCosminexus TP1/Message Control	TP1/Message Control
uCosminexus TP1/Message Control(64)	
uCosminexus TP1/Message Control/Tester	TP1/Message Control/ Tester
uCosminexus TP1/Message Queue	TP1/Message Queue
uCosminexus TP1/Message Queue(64)	
uCosminexus TP1/Message Queue - Access	TP1/Message Queue - Access
uCosminexus TP1/Message Queue - Access(64)	
uCosminexus TP1/Messaging	TP1/Messaging
uCosminexus TP1/Multi	TP1/Multi
uCosminexus TP1/NET/HDLC	TP1/NET/HDLC
uCosminexus TP1/NET/High Availability	TP1/NET/High Availability
uCosminexus TP1/NET/High Availability(64)	
uCosminexus TP1/NET/HSC	TP1/NET/HSC
uCosminexus TP1/NET/Library	TP1/NET/Library
uCosminexus TP1/NET/Library(64)	
uCosminexus TP1/NET/NCSB	TP1/NET/NCSB
uCosminexus TP1/NET/OSAS-NIF	TP1/NET/OSAS-NIF
uCosminexus TP1/NET/OSI-TP	TP1/NET/OSI-TP

Full name or meaning	Abbreviation
uCosminexus TP1/NET/Secondary Logical Unit - TypeP2	TP1/NET/Secondary Logical Unit - TypeP2
	TP1/NET/SLU - TypeP2
uCosminexus TP1/NET/TCP/IP	TP1/NET/TCP/IP
uCosminexus TP1/NET/TCP/IP(64)	
uCosminexus TP1/NET/User Datagram Protocol	TP1/NET/UDP
uCosminexus TP1/NET/User Agent	TP1/NET/User Agent
uCosminexus TP1/NET/X25	TP1/NET/X25
uCosminexus TP1/NET/X25-Extended	TP1/NET/ X25-Extended
uCosminexus TP1/NET/XMAP3	TP1/NET/XMAP3
uCosminexus TP1/Offline Tester	TP1/Offline Tester
uCosminexus TP1/Online Tester	TP1/Online Tester
uCosminexus TP1/Resource Manager Monitor	TP1/Resource Manager Monitor
uCosminexus TP1/Server Base	TP1/Server Base
uCosminexus TP1/Server Base(64)	
uCosminexus TP1/Shared Table Access	TP1/Shared Table Access
uCosminexus TP1/Web	TP1/Web
UNIX(R)	UNIX
Microsoft(R) Visual Studio(R) 2005 Professional Edition	Visual Studio
Microsoft(R) Visual Studio(R) 2005 Standard Edition	
Microsoft(R) Visual Studio(R) 2005 Team Edition for Software Architects	
Microsoft(R) Visual Studio(R) 2005 Team Edition for Software Developers	
Microsoft(R) Visual Studio(R) 2005 Team Edition for Software Testers	

Full name or meaning	Abbreviation		
Microsoft(R) Visual Studio(R) 2005 Team Suite			
Microsoft(R) Visual Studio(R) 2008 Professional Edition			
Microsoft(R) Visual Studio(R) 2008 Standard Edition			
Microsoft(R) Visual Studio(R) Team System 2008 Architecture Edition			
Microsoft(R) Visual Studio(R) Team System 2008 Development Edition			
Microsoft(R) Visual Studio(R) Team System 2008 Database Edition			
Microsoft(R) Visual Studio(R) Team System 2008 Team Suite			
Microsoft(R) Visual Studio(R) Team System 2008 Test Edition			
Microsoft(R) Windows(R) 2000 Advanced Server Operating System	Windows 2000		Windows
Microsoft(R) Windows(R) 2000 Datacenter Server Operating System			
Microsoft(R) Windows(R) 2000 Professional Operating System			
Microsoft(R) Windows(R) 2000 Server Operating System			
Microsoft(R) Windows(R) 7 Enterprise	Windows 7		
Microsoft(R) Windows(R) 7 Professional			
Microsoft(R) Windows(R) 7 Ultimate			
Microsoft(R) Windows Server(R) 2003, Datacenter Edition	Windows Server 2003	Windows Server 2003	
Microsoft(R) Windows Server(R) 2003, Enterprise Edition			
Microsoft(R) Windows Server(R) 2003, Standard Edition			

Full name or meaning	Abbreviation		
Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition	Windows Server 2003 R2		
Microsoft(R) Windows Server(R) 2003 R2, Standard Edition			
Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition	Windows Server 2003 R2 x64 Editions		
Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition			
Microsoft(R) Windows Server(R) 2003, Datacenter x64 Edition	Windows Server 2003 x64 Editions		
Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition			
Microsoft(R) Windows Server(R) 2003, Standard x64 Edition			
Microsoft(R) Windows Server(R) 2008 Datacenter (x86)	Windows Server 2008	Windows Server 2008	
Microsoft(R) Windows Server(R) 2008 Enterprise (x86)			
Microsoft(R) Windows Server(R) 2008 Standard (x86)			
Microsoft(R) Windows Server(R) 2008 R2, Datacenter Edition	Windows Server 2008 R2		
Microsoft(R) Windows Server(R) 2008 R2, Enterprise Edition			
Microsoft(R) Windows Server(R) 2008 R2, Standard Edition			
Microsoft(R) Windows Server(R) 2008 Datacenter (x64)	Windows Server 2008 x64 Editions		
Microsoft(R) Windows Server(R) 2008 Enterprise (x64)			
Microsoft(R) Windows Server(R) 2008 Standard (x64)			
Microsoft(R) Windows Vista(R) Business (x86)	Windows Vista	Windows Vista	
Microsoft(R) Windows Vista(R) Enterprise (x86)			
Microsoft(R) Windows Vista(R) Ultimate (x86)			

Full name or meaning	Abbreviation		
Microsoft(R) Windows Vista(R) Business (x64)	Windows Vista x64 Editions		
Microsoft(R) Windows Vista(R) Enterprise (x64)			
Microsoft(R) Windows Vista(R) Ultimate (x64)			
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP		

(2) Conventions: Acronyms

This manual also uses the following acronyms:

Acronym	Full name or meaning
API	Application Programming Interface
CPU	Central Processing Unit
CUP	Client User Program
DAM	Direct Access Method
DB	Database
DBA	Database Administrator
DBMS	Database Management System
DLL	Dynamic Link Library
EOF	End of File
GUI	Graphical User Interface
HA	High Availability
I/O	Input/Output
ID	Identifier
IP	Internet Protocol
ISAM	Indexed Sequential Access Method
IST	Internode Shared Table
J2EE	Java 2 Enterprise Edition
MCF	Message Control Facility

Acronym	Full name or meaning
MHP	Message Handling Program
MQI	Message Queue Interface
MSCS	Microsoft Cluster Service
MSDTC	Microsoft Distributed Transaction Coordinator
NIC	Network Interface Card
NTFS	New Technology File System
ODBC	Open DataBase Connectivity
OS	Operating System
OSI	Open Systems Interconnection
OSI TP	Open Systems Interconnection Transaction Processing
PRF	Performance
RM	Resource Manager
RPC	Remote Procedure Call
RTS	Real Time Statistic
SPP	Service Providing Program
SUP	Service Using Program
TAM	Table Access Method
TCP/IP	Transmission Control Protocol/Internet Protocol
TP	Transaction Processing
UAP	User Application Program
UOC	User Own Coding
WER	Windows Error Reporting
XA	Extended Architecture
XAR	Extended Architecture Resource

(3) Conventions: Differences between JIS and ASCII keyboards

The JIS and ASCII code maps assign different input characters to the codes shown below. The characters used in the code input examples in this manual (code examples)

are based on the JIS code map (Japanese keyboard).

Code	JIS keyboard	ASCII keyboard
$(5c)_{16}$	¥ (yen sign)	\ (backslash)
$(7e)_{16}$	‾ (overscore)	~ (swung dash)

(4) Conventions: Differences between upper-case and lower-case characters

OpenTP1 distinguishes between upper-case and lower-case characters (case sensitive). Make sure that you keep this distinction in mind when you specify command options or when you use character strings for command parameters in definition files. For example, the `-a` option is not the same as the `-A` option.

(5) Conventions: Descriptions of C language functions and COBOL UAPs

If there are no significant differences in a description with respect to C language functions and COBOL UAP creation programs, the C language functions are used in the description.

(6) Conventions: Environment variables

Windows environment variables are written as `%DCDIR%`. To use a Windows environment variable value, make sure that you type `%DCDIR%`, not `$DCDIR`.

(7) Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is $1{,}024^2$ bytes.
- 1 GB (gigabyte) is $1{,}024^3$ bytes.
- 1 TB (terabyte) is $1{,}024^4$ bytes.

(8) Conventions: Path name specifications

You must always include the drive letter when you specify an absolute path name. In a Windows environment, the colon (`:`) is used to separate the drive name from the folder names. Do not specify a colon in a path name anywhere other than following the drive letter.

Use a semicolon (`;`) to separate multiple path names.

(9) Conventions: Version numbers

The version numbers of Hitachi program products are usually written as two sets of two digits each, separated by a hyphen. For example:

- Version 1.00 (or 1.0) is written as 01-00.

- Version 2.05 is written as 02-05.
- Version 2.50 (or 2.5) is written as 02-50.
- Version 12.25 is written as 12-25.

The version number might be shown on the spine of a manual as *Ver. 2.00*, but the same version number would be written in the program as *02-00*.

Contents

Preface i

Intended readers i
Organization of this manual ii
Related publications iii
Conventions: Fonts and symbols....iv
Conventions: Other conventions used in this manual vii

1. Positioning of This Manual 1

1.1 Relationship to other OpenTP1 series manuals....2
1.2 How to use this manual 3

2. Description of Functions 7

2.1 List of functions....8
2.2 Functions for checking standard output and standard error 13
2.2.1 Standard output redirect facility....13
2.2.2 Console output facility for standard output redirect files....15
2.2.3 OpenTP1 console output facility....16
2.2.4 Relationships among system definitions....17
2.3 Permission control facility....18

3. Programming 19

3.1 List of code-level functions....20
3.2 Compiling and linking UAPs 33
3.2.1 Creating UAPs in C language 33
3.2.2 Creating UAPs for Visual Studio projects....35
3.2.3 Creating UAPs in COBOL2002....36
3.2.4 Creating UAPs by using Net Express 40
3.3 Programming notes specific to OpenTP1 for Windows....44
3.4 Sample programs....46
3.5 Handling of UAP shared libraries when using dynamic loading of service functions....48

4. Environment Setup 53

4.1 Environment setup procedure....54
4.2 Registering the OpenTP1 administrator 57
4.3 Creating OpenTP1 system definition information 58
4.3.1 Creating definition files....58
4.3.2 Changing definitions 58

4.3.3 Setting the runtime library path 58
4.4 Environment variable settings 60
4.5 Configuring an OpenTP1 file system 62
4.5.1 OpenTP1 file system 62
4.5.2 Configuring the OpenTP1 file system by applying the raw I/O facility 65
4.6 Settings for linking databases 68
4.6.1 Linking with HiRDB 68
4.6.2 Linking with Oracle 73
4.6.3 Linking with SQL Server 81
4.7 OS settings 88
4.8 MultiOpenTP1 settings 90

5. System Definitions 93

5.1 Lists of system definitions 94
5.2 Format of system definition explanations 150
5.3 Details of system definitions 151

6. Operations 171

6.1 Overview of OpenTP1 for Windows operations 172
6.2 Starting OpenTP1 173
6.2.1 Starting OpenTP1 from the Windows Services dialog box 173
6.2.2 Starting OpenTP1 with a command 174
6.2.3 Starting OpenTP1 with the GUI 175
6.2.4 Determining the startup mode 175
6.3 Terminating OpenTP1 177
6.3.1 Terminating OpenTP1 from the Windows Services dialog box 177
6.3.2 Terminating OpenTP1 with a command 177
6.3.3 Terminating OpenTP1 with the GUI 178
6.3.4 Notes about OpenTP1 termination 178
6.4 MultiOpenTP1 operations 179
6.5 System operations using scenario templates 180
6.5.1 Registering scenarios 180
6.5.2 Scale-out operation 180
6.5.3 Using scenario templates 182
6.5.4 Details of scenario templates 186
6.6 Troubleshooting 190
6.6.1 Error events and causes 190
6.6.2 Action to be taken in the event of an application exception 191
6.6.3 Action to be taken in the event of a shortage of ports (tuning the number of TCP/IP ports) 192
6.6.4 Using the information acquired in the event of a failure 193
6.7 GUI provided by OpenTP1 for Windows 195
6.7.1 GUI for environment setup 195
6.7.2 GUI for starting and terminating OpenTP1 205

7. Operation Commands 211

List of operation commands 212
Format of command explanations 230
Details of commands common to OpenTP1 for Windows and OpenTP1 for UNIX 231
dcpplist (displays product information)....232
dcrasget (acquires maintenance data)....233
fills (displays the contents of the OpenTP1 file system)....234
jnlcolc (collects journals required for file recovery) 235
jnlcopy (copies unload journal files) 236
jnledit (edits and outputs unload journal files or global archive unload journal files)..237
jnlmcst (outputs MCF statistics)....238
jnlrput (outputs records from unload journal files or global archive unload journal files) 239
jnlsort (sorts and merges unload journal files and global archive unload journal files in time order) 240
jnlstts (outputs statistical information) 241
jnlunlfg (unloads journal files) 242
rpcdump (outputs RPC trace information) 243
rpcmrg (merges RPC trace information) 244
trnlnkrm (registers resource managers)....245
trnmkobj (creates a transaction control object file)....247
Details of the commands specific to OpenTP1 for Windows....248
dcdls (checks the setup status of multiOpenTP1s)....249
dcmakecon (creates a multiOpenTP1 console) 251
dcsetupml (sets up a multiOpenTP1 or cancels setup of a multiOpenTP1) 253
njsmkdll (creates a transaction control DLL)....256
ntbstart (starts OpenTP1)....258
ntbtail (outputs the contents of a specified file to the console) 261
tp1console (outputs the standard output redirect file to the console)....263

Appendix 265

A. Lists of OpenTP1 Processes 266

Index 273

Chapter

1. Positioning of This Manual

We recommend that you read this manual in conjunction with other OpenTP1 series manuals that are relevant to your interests and purposes. This chapter explains the relationship between this manual and the other manuals in the OpenTP1 series, and explains how to use this manual.

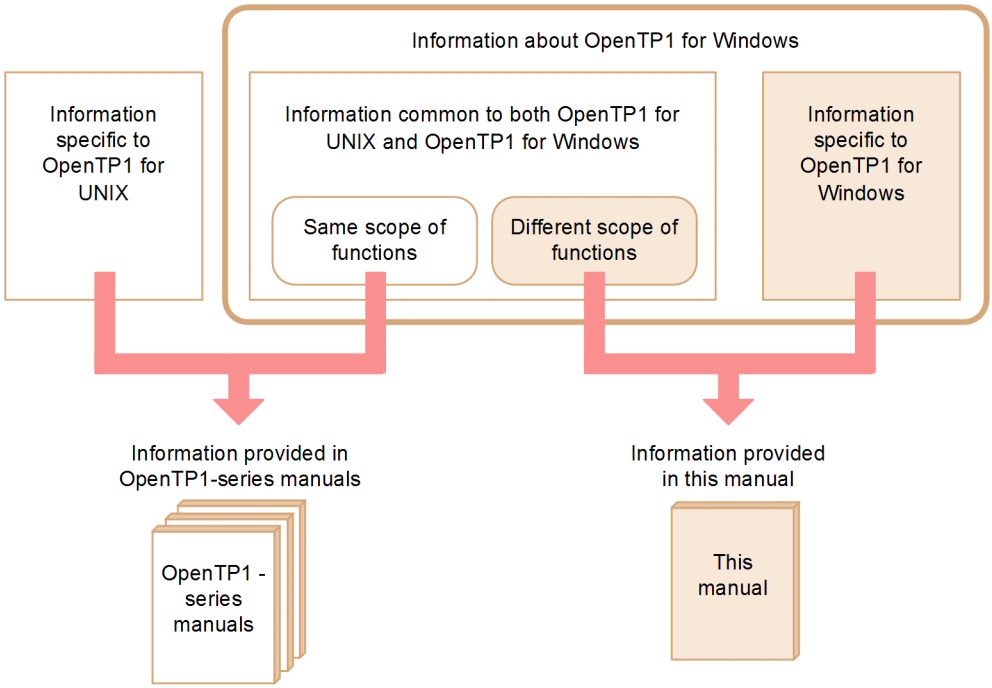
1.1 Relationship to other OpenTP1 series manuals
1.2 How to use this manual

1.1 Relationship to other OpenTP1 series manuals

This manual provides information specific to OpenTP1 for Windows, such as the OpenTP1 functions in Windows and the use of OpenTP1 to create programs in a Windows environment. It also explains the differences between OpenTP1 for Windows and OpenTP1 for UNIX. Information that is common to both OpenTP1 for Windows and OpenTP1 for UNIX is not included in this manual. For this reason, to use OpenTP1 for Windows, you must read this manual together with other appropriate manuals in the OpenTP1 series.

The following figure shows the relationship between this manual and the other manuals in the OpenTP1 series.

Figure 1-1: Relationship between this manual and the other OpenTP1 series manuals



1.2 How to use this manual

There are differences between OpenTP1 for Windows and OpenTP1 for UNIX in terms of the available functions, programming, environment setup, and system definition operands that are supported. We recommend that you use this manual and the other OpenTP1 series manuals as follows:

1. First, use this manual to determine whether there are differences in the OpenTP1 information between Windows and UNIX.

   See the lists provided in this manual starting in Chapter 2, such as the lists of program-level and code-level functions, to check for differences between OpenTP1 for Windows and OpenTP1 for UNIX.

2. Based on the result of this checking, use this manual or another manual in the OpenTP1 series, depending on which is appropriate.

   - This manual: Use this manual to obtain information specific to OpenTP1 for Windows and details about the differences with OpenTP1 for UNIX.
   - Other OpenTP1 series manuals: Use these manuals to obtain information common to both OpenTP1 for Windows and OpenTP1 for UNIX.

The manual to be used depends on the information about OpenTP1 for Windows that you want to obtain. The following figure shows the manual to be referenced, depending on the type of information about OpenTP1 for Windows that you are interested in.

Figure 1-2: Information about OpenTP1 for Windows and the manuals to be referenced

Information	Chapter in this manual	Other OpenTP1 series manual to be referenced
Overview of OpenTP1	—	*OpenTP1 Version 7 Description*
Program-level functions	Chapter 2	*OpenTP1 Version 7 Description*
Programming	Chapter 3	*OpenTP1 Version 7 Programming Guide*
Code-level functions	Chapter 3	*OpenTP1 Version 7 Programming Reference*#
Environment setup	Chapter 4	*OpenTP1 Version 7 Operation*
System definition	Chapter 5	*OpenTP1 Version 7 System Definition*
Operations	Chapter 6	*OpenTP1 Version 7 Operation*
Operation commands	Chapter 7	*OpenTP1 Version 7 Operation*
Messages	—	*OpenTP1 Version 7 Messages PartI/PartII*

Legend:

— : Not applicable

#: Indicates the following two manuals:
OpenTP1 Version 7 Programming Reference C Language
OpenTP1 Version 7 Programming Reference COBOL Language

We recommend that you read the indicated chapters in this manual as well as the other OpenTP1 series manuals, as appropriate to the information you desire, as indicated in this figure.

For example, to learn about the OpenTP1 functions, see both Chapter *2. Description of Functions* in this manual and the manual *OpenTP1 Version 7 Description*. Reference *2.1 List of functions* to determine whether there are differences between OpenTP1 for Windows and OpenTP1 for UNIX, and then see the corresponding section in Chapter 2 for details about the functions specific to OpenTP1 for Windows and the manual *OpenTP1 Version 7 Description* for the information common to both OpenTP1 for Windows and OpenTP1 for UNIX.

This manual does not explain the messages. For details about the messages, see the manual *OpenTP1 Version 7 Messages*.

Chapter

2. Description of Functions

The program-level functions that are available in OpenTP1 for Windows are different from those in OpenTP1 for UNIX. This chapter provides a list of program-level functions and explains the differences in these functions between OpenTP1 for Windows and OpenTP1 for UNIX. It also provides details of the functions that are specific to OpenTP1 for Windows.

For information common to both editions of OpenTP1, see the applicable OpenTP1 series manuals.

2.1 List of functions
2.2 Functions for checking standard output and standard error
2.3 Permission control facility

2.1 List of functions

The program-level functions provided by OpenTP1 for Windows are different from those provided by OpenTP1 for UNIX. If you are new to OpenTP1 for Windows, we recommend that you check the list below for the differences from OpenTP1 for UNIX.

For details about the functions that are common to both OpenTP1 for Windows and OpenTP1 for UNIX or the functions that are different between them, see the manual *OpenTP1 Version 7 Description*. For details about the functions specific to OpenTP1 for Windows, see the applicable sections in this chapter.

The following symbols are used in the list:

Legend:

S: Same (function is supported identically in OpenTP1 for Windows and OpenTP1 for UNIX)

W: Windows-specific (function is specific to OpenTP1 for Windows)

D: Differences (function has differences in OpenTP1 for Windows from OpenTP1 for UNIX or some limitations apply to OpenTP1 for Windows)

N: Not supported (definition is not supported for OpenTP1 for Windows)

--: Not applicable

The following table lists and describes the differences in functionality between OpenTP1 for Windows and OpenTP1 for UNIX.

Table 2-1: Differences in functionality between OpenTP1 for Windows and OpenTP1 for UNIX

Function	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Integrated system operations management JP1	D	The following system operation functions that are enabled by using JP1 products are not supported: • Distribution and asset management by JP1/Software Distribution • Network management by JP1/Cm2 • Registration of events into the JP1 event service of JP1/Base
UAP testing and debugging functions	N	The online and offline tester facilities and MCF online tester facility are not supported.

Function		Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Transaction control	Distributed transactions	S	--
	Global transactions	S	--
	Transaction commit and rollback	S	--
	Two-phase commit	S	--
	Transaction control using the TX interface	S	--
	Transaction control by the XA resource service	S	--
Communication in client/server mode	OpenTP1 remote procedure call communication	D	Service requests with domain qualification are not supported because `DCRPC_DOMAIN` is not supported for `flags` in the `dc_rpc_call` function.
	Optional functions for service information searches: • Global search facility • Service information prioritizing function	S	--
	OpenTP1 node management: • Start notification function • Node monitoring function	S	--
	XATMI interface communication	N	XATMI interface communication is not supported.
	TxRPC interface communication	N	TxRPC interface communication is not supported.

Function		Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Message control		D	The following functions are not supported: • User exit routine that determines inheritance of timer-start messages • MCF online tester • Specification as to whether acquisition of SPP log information OJ is required • High-speed MCF start processing when a memory queue is used for an input queue • Start of SPPs from MCF • MCF online command service
Application program scheduling	SPP scheduling	S	--
	MHP scheduling	S	--
	Process control • Multi-server load-balancing facility • Schedule priority • Inter-node load-balancing facility • Multi-scheduler facility	S	--
	Saving shared memory by sharing buffer area	N	The `scdbufgrp` schedule service definition command is not supported.
OpenTP1 client (TP1/Client)		S	--
Client/server-mode communication using OSI TP		N	The XATMI interface functions are used in client/server-mode communication using OSI TP. OpenTP1 for Windows does not support these functions because it does not support the XATMI interface.
Remote API facility		S	--
Dynamic loading of service functions		S	--
Functions that aids OpenTP1 operation	Locking resources	S	--

Function		Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Acquiring user journals	S	--
	Journal maintenance	D	Command specifications are different.
	Message log operation	D	Standard output is supported because the `logcat` command is supported, but output to `syslog` is not supported.
	Notification of message log information	S	--
	Control of a resource manager that is not provided by OpenTP1 (TP1/Resource Manager Monitor)	N	Resource manager control is not supported.
	Statistical information	D	Command specifications for statistical information are different. Facilities for handling MCF statistics (facilities for acquiring, outputting, and editing MCF statistics) are not supported (MCF statistics cannot be acquired).
	Real-time statistics service	S	--
System operation using scenario templates		D	Methods for using scenario templates are different.
System monitoring using audit logs		S	--
OpenTP1 file system		D	Operations involving access permissions for the OpenTP1 file system (OpenTP1 file protection) are not supported.
Functions for using multiple copies of OpenTP1	System switchover facility	D	OpenTP1 for Windows requries Hitachi HA Toolkit Extension, not an HA monitor. There are also the following differerences: • System configuration • Only `AUTO` can be specified in the `mode_conf` system environment definition operand. • The `dcstart` command is not supported.
	Multi-node (TP1/ Multi)	N	The multi-node function is not supported.

Function		Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	multiOpenTP1	D	Operating methods (operation commands) are different.
Structure of OpenTP1 processes		D	There are differences, such as some processes are specific to OpenTP1 for Windows, and the number of running processes differs. For details, see Appendix *A. List of OpenTP1 Processes*.
Functions specific to OpenTP1 for Windows	Standard output redirect facility	W	This function is specific to OpenTP1 for Windows. For details, see *2.2.1 Standard output redirect facility*.
	Console output facility for standard output redirect files	W	This function is specific to OpenTP1 for Windows. For details, see *2.2.2 Console output facility for standard output redirect files*.
	OpenTP1 console output facility	W	This function is specific to OpenTP1 for Windows. For details, see *2.2.3 OpenTP1 console output facility*.
	Permission control facility	W	This function is specific to OpenTP1 for Windows. For details, see *2.3 Permission control facility*.

Note:

For details about the functions common to both OpenTP1 for Windows and OpenTP1 for UNIX or differences between them, see the manual *OpenTP1 Version 7 Description* and check the function names provided in this table.

2.2 Functions for checking standard output and standard error

In OpenTP1 for Windows, you can output to files and to the console the information that is output by the system server and user servers to standard output and standard error (including the KFCA-*xxxxx* messages and all information directed to standard output by the users).

OpenTP1 provides the following facilities for checking standard output and standard error:

- Standard output redirect facility

  This facility outputs to a file the contents of standard output and standard error. It enables you to retain log information. You must open the file each time you want to check a file's contents.

- Console output facility for standard output redirect files

  This facility enables you to check in real time at the console the contents of a file output by the standard output redirect facility.

- OpenTP1 console output facility

  This facility outputs directly to the console the contents of standard output and standard error. The output information is not saved to a file.

If neither the standard output redirect facility nor the OpenTP1 console output facility is used, information that is output from a UAP by using a stream such as the `printf()` function is not displayed. If you want to check the information output by using a stream, you can use either the standard output redirect facility or the OpenTP1 console output facility.

The following subsections describe the individual facilities.

2.2.1 Standard output redirect facility

OpenTP1 for Windows can output directly to a file the contents of standard output and standard error. This function is called the *standard output redirect facility*, and it enables the user to save to a file the information that is output to standard output and standard error by the system server and user server (including the KFCA-*xxxxx* messages and all information specified for standard output by the user). You must open the file each time you want to check the file's contents.

This subsection explains the environment settings required in order to use the standard output redirect facility.

Reference note:

If you use the console output facility for standard output redirect files, you can check the output contents in real time at the console. For details about the console output facility for standard output redirect files, see *2.2.2 Console output facility for standard output redirect files*.

(1) Environment settings

To use the standard output redirect facility, you must specify the following system definition:

System environment definition

- `redirect_file` operand

  Specifies whether the contents of standard output and standard error are to be saved to a file.

- `redirect_file_name` operand

  Specifies the name of the file to which the contents of standard output and standard error are to be saved.

- `redirect_file_size` operand

  Specifies the maximum size of the file to which the contents of standard output and standard error are to be saved.

For details about these operands, see *System environment definition* in *5.3 Details of system definitions*.

Note that this facility cannot be used when you specify the `prc_port` and `rpc_port_base` operands in the system common definition. If these operands are specified, and `Y` is specified in the `redirect_file` operand, OpenTP1 outputs the `KFCA26520-W` message, ignores the `redirect_file` operand, and resumes the OpenTP1 startup operation. When the `KFCA26520-W` message is displayed, either delete the `prc_port` and `rpc_port_base` operands or delete the `redirect_file` operand.

(2) Notes

- If initialization of this facility fails, the `KFCA00715-E` message with reason code `200` is output to the Event Viewer. When this message is output, stop the OpenTP1 service and then restart OpenTP1.
- If write operations occur frequently during high workload periods, messages might be discarded.
- Standard output and standard error messages from processes running under OpenTP1 are output to a file via the `prctee` process. If the `prctee` process is shut down during online operation, standard output and standard error messages

from the processes running under OpenTP1 can no longer be output to a file. If this occurs, stop the OpenTP1 service and then restart OpenTP1.

- If the `redirect_file_name` operand is omitted or opening of the file specified in the `redirect_file_name` operand fails for a reason such as the specified folder being nonexistent, the following files are created as the destinations for standard output and standard error from a process running under OpenTP1:
  - `%DCDIR%\spool\prclog1`
  - `%DCDIR%\spool\prclog2`
- When this facility is used, any message in a batch file specified in the `user_command_online` system environment definition operand or in the executable file cannot be output to standard output or standard error. If a batch file or executable file that uses standard output and standard error is used, OpenTP1 issues the `KFCA01819-E` message and then shuts itself down. If you use a program that uses standard output and standard error, first execute `@echo off` within the batch file so that standard output and standard error will be ignored.
- If you use a stream such as the `printf()` function from within a UAP, we recommend that you flush the stream once the output operation has been completed. If the stream is not flushed, its correct contents might not be output to a file. Even when streams are used, this measure is not necessary with respect to the `DISPLAY` statement in a COBOL UAP, because the contents are output in real time.

2.2.2 Console output facility for standard output redirect files

You must open the file created by the standard output redirect facility each time you want to check its contents. If you use the *console output facility for standard output redirect files*, you can use the standard output redirect facility to check the contents of the file output in real time at the console. By using this facility together with the standard output redirect facility, you can save the output contents to a file and also check the contents in real time.

In order to use this facility, you must also use the standard output redirect facility. For details about the environment settings for the standard output redirect facility, see *2.2.1(1) Environment settings*.

You can start the console from a command prompt or from the Windows **Start** menu. These startup methods are explained below.

- Starting from a command prompt

  Start a command prompt and then execute the `tp1console` command or the `ntbtail -t p` command.

  For details about these commands, see *tp1console (outputs standard output redirect files to the console)* or *ntbtail (outputs the contents of a specified file to*

the console) in Chapter *7. Operation Commands*.

In a multiOpenTP1 environment, you must execute these commands from the console that was created by the `dcmakecon` command. For details about the `dcmakecon` command, see *dcmakecon (creates a multiOpenTP1 console)* in Chapter *7. Operation Commands*.

- Windows **Start** menu

  In an original OpenTP1 (that has been installed by the installer), you can start the console from the Windows **Start** menu. From the **Start** menu, choose **Programs**, **OpenTP1**, then **OpenTP1 Console**.

2.2.3 OpenTP1 console output facility

You can output to the console information that is output from the system server and user servers. This function is called the *OpenTP1 console output facility*. The information that can be output to the console by this facility is the same as for the files that are output by the standard output redirect facility. The difference from the standard output redirect facility is that this facility outputs the contents of standard output and standard error to the console rather than to a file. For this reason, the output contents are not saved.

This facility is not available when you start the OpenTP1 service from a user account or when you use an edition of Windows that supports the session 0 isolation feature (such as Windows Server 2008, Windows Vista, and Windows 7). Note that if you are using an edition of Windows that supports the session 0 isolation feature and you specify the environment settings for this facility, no error message will be output.

This subsection explains the environment setting required in order to use the OpenTP1 console output facility.

(1) Environment setting

To use the OpenTP1 console output facility, you must specify the following system definition:

System environment definition

- `console_output` operand

  Specifies whether the contents of standard output and standard error are to be output to the console.

For details about this operand, see *System environment definition* in *5.3 Details of system definitions*.

Note that you can use the OpenTP1 console output facility even when the `prc_port` and `rpc_port_base` operands are specified in the system common definition.

(2) Notes

- Make sure that the console used for standard output and standard error will never be closed. If the console is closed, the OpenTP1 system will shut down. If this occurs, some user and system servers might output the KFCA01820-E message and terminate abnormally with termination status 0xC000013A.
- You must not perform operations such as text selection and mouse clicks on the console to which standard output and standard error are being output. If such an operation is performed, display of the output contents might stop, message output might be delayed, and the OpenTP1 system might shut down.
- If analysis of the system environment definition results in an error, standard output and standard error are output to the Event Viewer.

2.2.4 Relationships among system definitions

The following table shows the relationships among the operands that are specified in order to use the function for checking standard output and standard error.

Table 2-2: Relationships among the operands that are specified in order to use the functions for checking standard output and standard error

Value of console_output operand	Specification of prc_port or rpc_port_base operand	Value of redirect_file operand	
		Y	N
Y	Yes	The OpenTP1 console output facility is effective.	The OpenTP1 console output facility is effective.
	No	The standard output redirect facility is effective.	
N	Yes	No output	No output
	No	The standard output redirect facility is effective.	

Legend:

Yes: Operand is specified

No: Operand is not specified

2.3 Permission control facility

In Windows, many permissions are granted by default to a user with Administrators group permissions. However, granting too many permissions might have adverse effects on the security of user servers.

OpenTP1 for Windows enables you to control the Windows permissions assigned to a user server. This function is called the *permission control facility*. You can use this facility to enhance the security of user servers.

If a user server is used to perform operations that require Windows permissions, you can also set permissions individually. For example, if you use the `dc_adm_call_command` function within a user server, and the process to be executed requires permissions, you must grant the permissions individually to that user server.

The following subsection explains the environment settings required in order to use the permission control facility.

(1) Environment settings

To use the permission control facility, you must specify the following system definition:

User service definition

- `process_privilege_restrict` operand

  Specifies whether the Windows permissions are to be restricted for the user server.

- `process_privilege_name` operand

  Specifies the name of the Windows permissions to be granted to the user server.

For details about these operands, see *User service definition* in *5.3 Details of system definitions*.

Chapter
3. Programming

The code-level functions that are available in OpenTP1 for Windows and the programming methods that are supported are different from those in OpenTP1 for UNIX. This chapter presents a list of code-level functions and explains the differences in these functions between OpenTP1 for Windows and OpenTP1 for UNIX. It also explains how to create programs that are specific to OpenTP1 for Windows.

For information common to both editions of OpenTP1, see the applicable OpenTP1 series manuals.

3.1 List of code-level functions
3.2 Compiling and linking UAPs
3.3 Programming notes specific to OpenTP1 for Windows
3.4 Sample programs
3.5 Handling of UAP shared libraries when using dynamic loading of service functions

3.1 List of code-level functions

The code-level functions that are available differ between OpenTP1 for Windows and OpenTP1 for UNIX. In addition, limitations apply to some functions, such as arguments and settings that are not supported. If you are new to OpenTP1 for Windows, we recommend that you check the list below for the differences from OpenTP1 for UNIX.

Once you have checked this list, see the applicable language's edition of the manual *OpenTP1 Version 7 Programming Reference* for details of the functions.

Note:

This list does not include the functions for message queuing (TP1/Message Queue).

The following symbols are used in the list:

Legend:

S: Same (function is supported identically in OpenTP1 for Windows and OpenTP1 for UNIX)

D: Differences (function has differences in OpenTP1 for Windows from OpenTP1 for UNIX or some limitations apply to OpenTP1 for Windows)

N: Not supported (function is not supported for OpenTP1 for Windows)

--: Not applicable

The following table lists and describes differences in functionality between OpenTP1 for Windows and OpenTP1 for UNIX.

Table 3-1: Differences in functionality between OpenTP1 for Windows and OpenTP1 for UNIX

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Remote procedure call	Starting an application program	`dc_rpc_open` `CBLDCRPC('OPEN  ')`	S	--
	Starting an SPP service	`dc_rpc_mainloop` `CBLDCRSV('MAINLOOP')`	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Requesting a remote service	dc_rpc_call CBLDCRPC('CALL ')	D	The DCRPC_DOMAIN setting in flags is not supported.
	Invoking a remote service with a communication destination specified	dc_rpc_call_to --	S	--
	Receiving processing results in asynchronous mode	dc_rpc_poll_any_replies CBLDCRPC('POLLANYR')	S	--
	Acquiring a descriptor of an asynchronous-response RPC request resulting in an error	dc_rpc_get_error_descriptor CBLDCRPC('GETERDES')	S	--
	Rejecting reception of processing results	dc_rpc_discard_further_replies CBLDCRPC('DISCARDF')	S	--
	Discarding a specific response message of an asynchronous-response RPC	dc_rpc_discard_specific_reply CBLDCRPC('DISCARDS')	S	--
	Retrying a service function	dc_rpc_service_retry CBLDCRPC('SVRETRY')	S	--
	Setting a schedule priority for service requests	dc_rpc_set_service_prio CBLDCRPC('SETSVPRI')	S	--
	Referencing a schedule priority for service requests	dc_rpc_get_service_prio CBLDCRPC('GETSVPRI')	S	--
	Referencing the service response wait time	dc_rpc_get_watch_time CBLDCRPC('GETWATCH')	S	--
	Updating the service response wait time	dc_rpc_set_watch_time CBLDCRPC('SETWATCH')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Acquiring a client UAP's node address	dc_rpc_get_callers_address CBLDCRPC('GETCLADR')	S	--
	Acquiring a gateway node address	dc_rpc_get_gateway_address CBLDCRPC('GETGWADR')	S	--
	Unilaterally reporting to a CUP	dc_rpc_cltsend CBLDCRPC('CLTSEND ')	S	--
	Terminating an application program	dc_rpc_close CBLDCRPC('CLOSE ')	S	--
Remote API facility	Establishing a connection with a RAP-processing listener	dc_rap_connect CBLDCRAP('CONNECT ') CBLDCRAP('CONNECTX')	S	--
	Disconnecting from a RAP-processing listener	dc_rap_disconnect CBLDCRAP('DISCNCT ')	S	--
Transaction control	Starting a transaction	dc_trn_begin CBLDCTRN('BEGIN ')	S	--
	Committing in chained mode	dc_trn_chained_commit CBLDCTRN('C-COMMIT')	S	--
	Rolling back in chained mode	dc_trn_chained_rollback CBLDCTRN('C-ROLL ')	S	--
	Committing in unchained mode	dc_trn_unchained_commit CBLDCTRN('U-COMMIT')	S	--
	Rolling back in unchained mode	dc_trn_unchained_rollback CBLDCTRN('U-ROLL ')	S	--
	Reporting information about the current transaction	dc_trn_info CBLDCTRN('INFO ')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
System operations management	Executing operation commands	`dc_adm_call_command` `CBLDCADM('COMMAND ')`	D	• The `DCADM_DELAY` setting in `flags` is not supported. • If the specified character string contains `/` or `$`, OpenTP1 performs internally a specific character string conversion processing on the character string. If you want to ignore OpenTP1's character string conversion processing and execute the command exactly as specified, specify the following environment variable in the user service definition or user service default definition: `putenv DCADMCONVSEP 1` • OpenTP1 does not support asynchronous execution of a command when `&` is specified at the end of the character string.

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Reporting completion of user server start processing	dc_adm_complete CBLDCADM('COMPLETE')	S	--
	Reporting user server status	dc_adm_status CBLDCADM('STATUS ')	S	--
Output of audit log	Outputting an audit log	dc_log_audit_print CBLDCADT('PRINT ')	S	--
Output of message log	Outputting a message log	dc_logprint CBLDCLOG('PRINT ')	D	color settings are not supported.
Acquisition of a user journal	Acquiring a user journal	dc_jnl_ujput CBLDCJNL('UJPUT ')	S	--
Editing journal data	Closing a jnlrput output file	-- CBLDCJUP('CLOSERPT')	S	--
	Opening a jnlrput output file	-- CBLDCJUP('OPENRPT ')	S	--
	Importing journal data from a jnlrput output file	-- CBLDCJUP('RDGETRPT')	S	--
Message exchange	Opening an MCF environment	dc_mcf_open CBLDCMCF('OPEN ')	S	--
	Starting an MHP service	dc_mcf_mainloop CBLDCMCF('MAINLOOP')	S	--
	Receiving messages	dc_mcf_receive CBLDCMCF('RECEIVE ')	S	--
	Sending response messages	dc_mcf_reply CBLDCMCF('REPLY ')	S	--
	Sending messages	dc_mcf_send CBLDCMCF('SEND ')	S	--
	Resending messages	dc_mcf_resend CBLDCMCF('RESEND ')	S	--
	Receiving synchronous messages	dc_mcf_recvsync CBLDCMCF('RECVSYNC')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Sending synchronous messages	dc_mcf_sendsync CBLDCMCF('SENDSYNC')	S	--
	Exchanging synchronous messages	dc_mcf_sendrecv CBLDCMCF('SENDRECV')	S	--
	Accepting temporarily-stored data	dc_mcf_tempget CBLDCMCF('TEMPGET ')	S	--
	Updating temporarily-stored data	dc_mcf_tempput CBLDCMCF('TEMPPUT ')	S	--
	Ending continuous inquiry responses	dc_mcf_contend CBLDCMCF('CONTEND ')	S	--
	Activating an application program	dc_mcf_execap CBLDCMCF('EXECAP ')	S	--
	Reporting application information	dc_mcf_ap_info CBLDCMCF('APINFO ')	S	--
	Reporting application information to a user exit routine	dc_mcf_ap_info_uoc --	S	--
	Setting user timer monitoring	dc_mcf_timer_set CBLDCMCF('TIMERSET')	S	--
	Canceling user timer monitoring	dc_mcf_timer_cancel CBLDCMCF('TIMERCAN')	S	--
	Committing MHP	dc_mcf_commit CBLDCMCF('COMMIT ')	S	--
	Rolling back MHP	dc_mcf_rollback CBLDCMCF('ROLLBACK')	S	--
	Closing MCF environment	dc_mcf_close CBLDCMCF('CLOSE ')	S	--
	Acquiring MCF communication service statuses	dc_mcf_tlscom CBLDCMCF('TLSCOM ')	S	--
	Acquiring the connection status	dc_mcf_tlscn CBLDCMCF('TLSCN ')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Establishing a connection	dc_mcf_tactcn CBLDCMCF('TACTCN ')	S	--
	Releasing a connection	dc_mcf_tdctcn CBLDCMCF('TDCTCN ')	S	--
	Acquiring the acceptance status of a server-type connection establishment request	dc_mcf_tlsln CBLDCMCF('TLSLN ')	S	--
	Starting acceptance of a server-type connection establishment request	dc_mcf_tonln CBLDCMCF('TONLN ')	S	--
	Ending acceptance of a server-type connection establishment request	dc_mcf_tofln CBLDCMCF('TOFLN ')	S	--
	Deleting an application timer start requests	dc_mcf_adltap CBLDCMCF('ADLTAP ')	S	--
	Acquiring a logical terminal status	dc_mcf_tlsle CBLDCMCF('TLSLE ')	S	--
	Shutting down a logical terminal	dc_mcf_tdctle CBLDCMCF('TDCTLE ')	S	--
	Releasing a logical terminal from shutdown status	dc_mcf_tactle CBLDCMCF('TACTLE ')	S	--
	Deleting a logical terminal output queue	dc_mcf_tdlqle CBLDCMCF('TDLQLE ')	S	--
DAM file service	Opening a logical file	dc_dam_open CBLDCDAM('DCDAMSVC','OPEN')	S	--
	Inputting logical file blocks	dc_dam_read CBLDCDAM('DCDAMSVC','READ')	S	--
	Updating logical file blocks	dc_dam_rewrite CBLDCDAM('DCDAMSVC','REWT')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Outputting logical file blocks	dc_dam_write CBLDCDAM('DCDAMSVC','WRIT')	S	--
	Closing a logical file	dc_dam_close CBLDCDAM('DCDAMSVC','CLOS')	S	--
	Shutting down a logical file	dc_dam_hold CBLDCDAM('DCDAMSVC','HOLD')	S	--
	Releasing a logical file from shutdown status	dc_dam_release CBLDCDAM('DCDAMSVC','RLES')	S	--
	Referencing the status of a logical file	dc_dam_status CBLDCDAM('DCDAMSVC','STAT')	S	--
	Starting an unrecoverable DAM file	dc_dam_start CBLDCDAM('DCDAMSVC','STRT')	S	--
	Ending an unrecoverable DAM file	dc_dam_end CBLDCDAM('DCDAMSVC','END ')	S	--
	Allocating a physical file	dc_dam_create CBLDCDMB('DCDAMINT','CRAT')	S	--
	Opening a physical file	dc_dam_iopen CBLDCDMB('DCDAMINT','OPEN')	S	--
	Inputting physical file blocks	dc_dam_get CBLDCDMB('DCDAMINT','GET ')	S	--
	Outputting physical file blocks	dc_dam_put CBLDCDMB('DCDAMINT','PUT ')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Searching physical file blocks	dc_dam_bseek CBLDCDMB('DCDAMINT','BSEK')	S	--
	Directly inputting physical file blocks	dc_dam_dget CBLDCDMB('DCDAMINT','DGET')	S	--
	Directly outputting physical file blocks	dc_dam_dput CBLDCDMB('DCDAMINT','DPUT')	S	--
	Closing a physical file	dc_dam_iclose CBLDCDMB('DCDAMINT','CLOS')	S	--
TAM file service	Opening a TAM table	dc_tam_open --	S	--
	Inputting TAM table records	dc_tam_read CBLDCTAM('FxxR')('FxxU')	S	--
	Updating on the assumption that TAM table records have been input	dc_tam_rewrite CBLDCTAM('MFY ')('MFYS')('STR ')	S	--
	Updating/adding TAM table records	dc_tam_write CBLDCTAM('MFY ')('MFYS')('STR ')	S	--
	Deleting TAM table records	dc_tam_delete CBLDCTAM('ERS ')('ERSR')	S	--
	Canceling input of TAM table records	dc_tam_read_cancel --	S	--
	Acquiring TAM table status	dc_tam_get_inf CBLDCTAM('GST ')	S	--
	Acquiring information about a TAM table	dc_tam_status CBLDCTAM('INFO')	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Closing a TAM table	dc_tam_close --	S	--
IST service	Opening an IST	dc_ist_open CBLDCIST('DCISTSVC','OPEN')	N	The IST service is not supported.
	Inputting IST records	dc_ist_read CBLDCIST('DCISTSVC','READ')	N	
	Outputting IST records	dc_ist_write CBLDCIST('DCISTSVC','WRIT')	N	
	Closing an IST	dc_ist_close CBLDCIST('DCISTSVC','CLOS')	N	
Locking a resource	Locking resources	dc_lck_get CBLDCLCK('GET ')	S	--
	Releasing all resources from a lock	dc_lck_release_all CBLDCLCK('RELALL ')	S	--
	Releasing a lock with a resource name specified	dc_lck_release_byname CBLDCLCK('RELNAME ')	S	--
XATMI interface	Calling a request/response service and receiving a response	tpcall() TPCALL	N	The XATMI interface is not supported.
	Calling a request/response service	tpacall() TPACALL	N	
	Receiving an asynchronous response from a request/response service	tpgetrply() TPGETRPLY	N	
	Canceling a request/response service	tpcancel() TPCANCEL	N	
	Connecting to an interactive service	tpconnect() TPCONNECT	N	

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Disconnecting from an interactive service	tpdiscon() TPDISCON	N	
	Receiving messages from an interactive service	tprecv() TPRECV	N	
	Sending messages to an interactive service	tpsend() TPSEND	N	
	Allocating a typed buffer	tpalloc() --	N	
	Releasing a typed buffer	tpfree() --	N	
	Changing a typed buffer's size	tprealloc() --	N	
	Acquiring information about a typed buffer	tptypes() --	N	
	Advertising a service name	tpadvertise() TPADVERTISE	N	
	Canceling advertisement of a service name	tpunadvertise() TPUNADVERTISE	N	
	Template for service functions	tpservice() TPSVCSTART	N	
	Returning from a service function	tpreturn() TPRETURN	N	
TX interface	Starting a transaction	tx_begin() TXBEGIN	S	--
	Committing a transaction	tx_commit() TXCOMMIT	S	--
	Returning information about the current transaction	tx_info() TXINFORM	S	--

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Opening a set of resource managers	tx_open() TXOPEN	S	--
	Rolling back a transaction	tx_rollback() TXROLLBACK	S	--
	Closing a set of resource managers	tx_close() TXCLOSE	S	--
	Setting the commit_return characteristic	tx_set_commit_return() TXSETCOMMITRET	S	--
	Setting the transaction_control characteristic	tx_set_transaction_control() TXSETTRANCTL	S	--
	Setting the transaction_timeout characteristic	tx_set_transaction_timeout() TXSETTIMEOUT	S	--
Multi-node	Starting acquisition of OpenTP1 node status	dc_adm_get_nd_status_begin --	N	The multi-node (TP1/Multi) is not supported.
	Acquiring the status of OpenTP1 nodes	dc_adm_get_nd_status_next --	N	
	Acquiring the status of a specified OpenTP1 node	dc_adm_get_nd_status --	N	
	Terminating the acquisition of OpenTP1 node status	dc_adm_get_nd_status_done --	N	
	Starting the acquisition of node identifiers	dc_adm_get_nodeconf_begin --	N	
	Acquiring node identifiers	dc_adm_get_nodeconf_next --	N	

Classification/purpose		Function name#	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Terminating acquisition of node identifiers	dc_adm_get_nodeconf_done --	N	
	Acquiring a specified node identifier	dc_adm_get_node_id --	N	
	Starting acquisition of user server statuses	dc_adm_get_sv_status_begin --	N	
	Acquiring the status of user servers	dc_adm_get_sv_status_next --	N	
	Acquiring the status of a specified user server	dc_adm_get_sv_status --	N	
	Terminating acquisition of user server statuses	dc_adm_get_sv_status_done --	N	
Management of online testers	Reporting the test status of a user server	dc_uto_test_status CBLDCUTO('T-STATUS')	N	The online tester facility (TP1/Online Tester) is not supported.
Performance verification trace	Acquiring user-specific performance verification traces	dc_prf_utrace_put CBLDCPRF('PRFPUT ')	S	--
	Reporting the sequential number for an acquired performance verification trace	dc_prf_get_trace_num CBLDCPRF('PRFGETN ')	S	--
Real-time statistics service	Acquiring real-time statistics at any interval	dc_rts_utrace_put CBLDCRTS('RTSPUT ')	S	--

#

The upper row shows the C language function name, and the lower row shows the COBOL UAP creation program.

3.2 Compiling and linking UAPs

This section provides and explains examples of the files and makefiles that are required in each development environment to compile and link UAPs.

For details about how to compile and link UAPs, see the applicable language's edition of the manual *OpenTP1 Version 7 Programming Reference*.

There are some points specific to OpenTP1 for Windows that must be noted before you begin programming. Check the information provided in *3.3 Programming notes specific to OpenTP1 for Windows* before you start compiling and linking UAPs.

3.2.1 Creating UAPs in C language

This subsection explains UAP creation in C language.

(1) Preparing a manifest file

You need a manifest file in order to run a program created in Visual Studio. Either embed the manifest file in the program or copy the manifest file to the folder where the executable file is located.

Reference note:

A manifest file contains information about the libraries (DLLs) that are required in order to run executable files (such as `.exe` and `.dll`). A manifest file is created automatically when an executable file is created. This is a mandatory file for running an executable file.

(2) Installing C runtime libraries

Because TP1/Server Base has been created by Visual Studio, you must install C runtime libraries to run TP1/Server Base.

For details about the C runtime libraries that must be installed, see the Release Notes.

(3) Example of a makefile (C language: SPP)

This subsection shows an example of a makefile needed for creating an SPP.

(a) Names of source files

The following shows the names of the source files that are used in the example of a makefile.

- Names of the source programs of the UAP created in C language
  - `exmain.c` (main function)
  - `exsv1.c` (service function 1)

- exsv2.c (service function 2)

■ Name of the RPC interface definition file: ex.def

(b) Example of a makefile

The following shows an example of a makefile:

```
NODEBUG = 1
!include <ntwin32.mak>
# TP1TEST Make File
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib

all:TP1TEST.exe
# Compile
exmain.obj:exmain.c
         $(cc) $(cflags) $(cvarsdll) $*.c
exsv1.obj:exsv1.c
         $(cc) $(cflags) $(cvarsdll) $*.c
exsv2.obj:exsv2.c
         $(cc) $(cflags) $(cvarsdll) $*.c
ex_sstb.obj:ex_sstb.c
         $(cc) $(cflags) $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
         $(DCDIR)\bin\stbmake $?
# Link
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj
         $(link) $(conflags) -out:$*.exe $** \
         [$(DCDIR)\spool\trnrmcmd\userobj\dctrninf.obj \]#3
         $(TP1LIB) $(SYSLIB)
         [mt -manifest $@.manifest -outputresource:$@;1]#4
```

#1

Specify libdam.lib only if you use TP1/FS/Direct Access.

#2

Specify libtam.lib only if you use TP1/FS/Table Access.

#3

Specify $(DCDIR)\spool\trnrmcmd\userobj\dctrninf.obj only if you use the transaction facility. For dctrninf.obj, specify the name of the object file for transaction control that was created by the trnmkobj command. For details about the trnmkobj command, see Chapter *7. Operation Commands*.

#4

Delete this row if you do not embed the manifest file.

(4) Example of a makefile (C language: MHP)

This subsection shows an example of a makefile needed for creating an MHP.

```
NODEBUG = 1
!include <ntwin32.mak>
# makefile for making UAP(C Language)

# command
STBMAKE           =       $(DCDIR)\bin\stbmake.exe

# stub name
STBMAKE_MHP_SRC   =       mhp_sstb

# user server object
MHP_SRC           =       mhpmain.obj mhp_sv.obj

# include file
INCDIR            =       -I.                 \
                          -I $(DCDIR)\include

# common object
SH_OBJ            =       $(DCDIR)\spool\trnrmcmd\userobj\mcf_sw.obj

# common library
LIBNAME           =       $(DCDIR)\lib\libmcf.lib \
                          $(DCDIR)\lib\libmnet.lib \
                          $(DCDIR)\lib\libbetran.lib

# compiler options
CC                =       $(cc)
CCOPT             =       -nologo $(cflags) $(cvarsdll)
CONFIG            =       -c
CFLAGS            =       $(CCOPT) $(CONFIG) $(INCDIR)

# linker options
LK                =       $(link)
LNKFLG            =       $(conflags) /NODEFAULTLIB
LINKAGE           =       $(conlibsdll) $(LIBNAME) msvcrt.lib

.c.obj:
       $(CC) $*.c $(CFLAGS)

all : $(SH_OBJ) tcpmhp.exe

$(STBMAKE_MHP_SRC).c :
        $(STBMAKE) -s $(STBMAKE_MHP_SRC).c $(STBMAKE_MHP_SRC).def

tcpmhp.exe : $(STBMAKE_MHP_SRC).obj $(MHP_SRC)
        $(LK) $(LNKFLG) $** $(LINKAGE) $(SH_OBJ) -out:$@
       !if exist $@.manifest mt.exe -nologo -manifest $@.manifest -outputresource:$@;1
        !if exist $@.manifest del $@.manifest

$(SH_OBJ) :
        $(DCDIR)\bin\trnmkobj -o mcf_sw -R OpenTP1_MCF
```

3.2.2 Creating UAPs for Visual Studio projects

This subsection explains creation of UAPs for Visual Studio projects. The example presented here is for Visual Studio 2008.

(1) Project type

The project type to be specified is `Win32 project`.

(2) Files composing the project

The files to be specified include the application's source files that have been created as the source files composing the project, and the stub source file.

(3) Project settings

The following table lists and describes the Visual Studio project settings.

Table 3-2: Visual Studio project settings

Category	Item		Setting
C/C++	General	Additional include directory	*OpenTP1-installation-folder*`\include`
	Code generation	Run-time libraries	Multi-threaded DLL (`/MD`)
	Precompiled header	Create and use a precompiled header	Precompiled headers are not used
Linker	General	Additional library directory	*OpenTP1-installation-folder*`\lib`
	Input	Additional dependent files	• `msvcrt.lib` • `libbetran.lib`
		Ignore all default libraries	Yes (`/NODEFAULTLIB`)
	System	Subsystem	Console (`/SUBSYSTEM:CONSOLE`)

3.2.3 Creating UAPs in COBOL2002

This subsection explains UAP creation in COBOL2002.

(1) Specifying commands (COBOL2002)

This subsection explains the commands that are specified in the following cases:

(a) Using the trnmkobj command to create transaction control objects

If you use the `trnmkobj` command to create transaction control objects, make sure that you specify the `-C "/Zl"` option.

(b) Using the default transaction control object file (dc_trn_allrm.obj)

If you use the default transaction control object file (`dc_trn_allrm.obj`) that is created by using the `trnlnkrm` command, execute the `trnlnkrm` command with the `-C "/Zl"` option specified and re-create the object.

(c) Compiling the stub file

If you compile the C language source file created by `stbmake`, make sure that you

specify the "/Zl" compiler option.

(2) Example of a makefile (COBOL2002: SPP)

This subsection shows an example of a makefile needed for creating an SPP.

(a) Names of source files

The following are the source files that are used in the example of a makefile:

- Main program of the UAP created in COBOL2002: main.cbl
- Service program 1 of the UAP created in COBOL2002: sev1.cbl
- Service program 2 of the UAP created in COBOL2002: sev2.cbl
- RPC interface definition file: ex.def

(b) Example of a makefile

The following shows an example of a makefile:

```
!include <ntwin32.mak>
# TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
# Compile
main.obj:main.cbl
         $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
         $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
         $(CCBL) $(CBLFG2) sev2.cbl
ex_sstb.obj:ex_sstb.c
         [$(cc) $(cflags) /Zl $(cvarsdll) $*.c]#3
# Create a stub source file
ex_sstb.c:ex.def
         $(DCDIR)\bin\stbmake $?
# Link
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj
         $(LK) -OutputFile $*.exe $** \
         [$(DCDIR)\spool\trnrmcmd\userobj\dctrninf.obj \]#4
         $(TP1LIB) $(SYSLIB)
```

#1

Specify libdam.lib only if you use TP1/FS/Direct Access.

#2

Specify libtam.lib only if you use TP1/FS/Table Access.

#3

For `ex_sstb.obj`, specify the `/Zl` compiler option.

#4

Specify `$(DCDIR)\spool\trnrmcmd\userobj\dctrninf.obj` only if you use the transaction facility. For `dctrninf.obj`, specify the name of the object file for transaction control that was created by the `trnmkobj` command. If you execute the `trnmkobj` command, specify the `-C "/Zl"` option so that information about the manifest file is not embedded in the object. For details about the `trnmkobj` command, see Chapter *7. Operation Commands*.

(3) Example of a makefile (COBOL2002: MHP)

This subsection shows an example of a makefile needed for creating an MHP.

```
!include <ntwin32.mak>
# makefile for making UAP(COBOL Language)

# COBOL2002 install directory
COBOL2002         = C:\Progra~1\HITACHI\COBOL2002

# command
STBMAKE           = $(DCDIR)\bin\stbmake.exe

# user server object
MHP_SRC           = CBLMHPM.obj TCPMHPSV.obj

# include file
INCDIR            =       -I.                 \
                          -I $(DCDIR)\include

# common object
SH_OBJ            =       $(DCDIR)\spool\trnrmcmd\userobj\mcf_sw.obj

# common library
LIBNAME           =       $(DCDIR)\lib\libmcf.lib \
                          $(DCDIR)\lib\libmnet.lib \
                          $(DCDIR)\lib\libbetran.lib

# COBOL2002 library
CBLLIBNAME        =       $(COBOL2002)\lib\cbl2k_32.lib \
                          $(COBOL2002)\lib\cbl2klc.lib \
                          $(COBOL2002)\lib\cbl2kdl.lib
```

```
# compiler options
CC                =       cl
CCBL              =       $(COBOL2002)\bin\ccbl2002
CCOPT             =       -nologo $(cflags) $(cvarsdll)
CONFIG            =       -c -Zl
CFLAGS            =       $(CCOPT) $(CONFIG) $(INCDIR)
CBLFG1            =       -Compile,Nolink -Comp5 -Lib,CUI -OpenTP1 -Main,System
CBLFG2            =       -Compile,Nolink -Comp5 -Lib,CUI -OpenTP1

# linker options
LK                =       ccbl2002 -Lib,CUI
LNKFLG            =       $(conflags) /NODEFAULTLIB
LINKAGE           =       $(conlibsdll) $(LIBNAME) $(CBLLIBNAME) msvcrt.lib

all : $(SH_OBJ) tcpmhp.exe

.SUFFIXES : .cbl

# compile
.c.obj   :
                $(CC) $(CFLAGS) $*.c

.cbl.obj :
                $(CCBL) $(CBLFG2) $*.cbl

TCPMHPSV.obj :
                $(CCBL) $(CBLFG2) TCPMHPSV.cbl

CBLMHPM.obj :
                $(CCBL) $(CBLFG1) cblmhpm.cbl

# stub, creating source file
cblmhp_sstb.c :
                $(STBMAKE) cblmhp.def

# link
tcpmhp.exe : $(MHP_SRC) cblmhp_sstb.obj
                $(LK) -OutputFile $*.exe $** $(LINKAGE) $(SH_OBJ)
                !if exist $@.manifest mt.exe -nologo -manifest $@.manifest
-outputresource:$@;1
                !if exist $@.manifest del $@.manifest

$(SH_OBJ) :
                $(DCDIR)\bin\trnmkobj -o mcf_sw -R OpenTP1_MCF -C "/Zl"
```

(4) COBOL2002 Development Manager settings

Specify the COBOL2002 Development Manager settings as follows:

- Compiler option setting

  Specify **CUI: Link with CUI run time libraries**.

- Linker option setting

  Specify `libbetran.lib` as an import or user-specified library, and specify a stub object file.

(5) Environment settings when Visual Studio is also used (version earlier than COBOL2002 02-00)

COBOL2002 with a version earlier than 02-00 is created by using Visual Studio .NET 2002. An error results if you use the `ccbl2002` command and the linker (link) included in COBOL2002 to link an object created by using Visual Studio (Visual Studio 2005 and Visual Studio 2008). In order to avoid such an error, you must specify the environment settings according to the following rules and then link the program:

- In the `PATH` and `LIB` environment variables, specify the path required in COBOL2002 before the path required in Visual Studio.
- If you compile stubs for SPP, specify the `/Zl` option.
- Create as a DLL a program that has been created in C language and that is not a stub for SPP.

For COBOL2002 02-00 or later, this error will not occur because COBOL2002 is created by using Visual Studio 2005.

3.2.4 Creating UAPs by using Net Express

This subsection explains UAP creation using Net Express.

(1) Specifying commands (Net Express)

This subsection explains the commands that are specified in the following cases:

(a) Creating transaction control objects by using the trnmkobj command

If you use the `trnmkobj` command to create transaction control objects, make sure that you specify the `-C "/Zl"` option.

(b) Using the default transaction control object file (dc_trn_allrm.obj)

If you use the default transaction control object file (`dc_trn_allrm.obj`) that is created by using the `trnlnkrm` command, execute the `trnlnkrm` command with the `-C "/Zl"` option specified and re-create the object.

(c) Compiling the stub file

If you compile the C language source file created by `stbmake`, make sure that you specify the `"/Zl"` compiler option.

(2) Example of a makefile (Net Express: SPP)

This subsection shows an example of a makefile needed for creating an SPP.

(a) Names of source files

The following shows the names of the source files that are used in the example of a makefile:

- Main program of the UAP created in Net Express: `main.cbl`

- Service program 1 of the UAP created in Net Express: `sev1.cbl`
- Service program 2 of the UAP created in Net Express: `sev2.cbl`
- RPC interface definition file: `ex.def`

(b) Example of a makefile

The following shows an example of a makefile:

```
!include <ntwin32.mak>
# TP1TEST Make File
CBLFG = /LITLINK /OBJ
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib user32.lib
CBL_LIB = cblrtss.lib cbllds.obj adis.obj adisinit.obj adiskey.obj mffh.obj
COBOL = cobol

all:TP1TEST.exe
# Compile
main.obj:main.cbl
         $(COBOL) $* $(CBLFG);
sev1.obj:sev1.cbl
         $(COBOL) $* $(CBLFG);
sev2.obj:sev2.cbl
         $(COBOL) $* $(CBLFG);
ex_sstb.obj:ex_sstb.c
         [$(cc) $(cflags) /Zl $(cvarsdll) $*.c]#3
# Create a stub source file
ex_sstb.c:ex.def
         $(DCDIR)\bin\stbmake $?
# Link
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj
         $(link) $(conflags) -out:$*.exe $** \
         [$(DCDIR)\spool\trnrmcmd\userobj\dctrninf.obj \]#4
         $(TP1LIB) $(SYSLIB) $(CBL_LIB)
```

#1

Specify `libdam.lib` only if you use TP1/FS/Direct Access.

#2

Specify `libtam.lib` only if you use TP1/FS/Table Access.

#3

For `ex_sstb.obj`, specify the `/Zl` compiler option.

#4

Specify `$(DCDIR)\spool\trnrmcmd\userobj\dctrninf.obj` only if you use the transaction facility. For `dctrninf.obj`, specify the name of object file for transaction control that was created by using the `trnmkobj` command. If you execute the `trnmkobj` command, specify the `-C "/Zl"` option so that information about the manifest file is not embedded in the object. For details

about the `trnmkobj` command, see Chapter *7. Operation Commands*.

(3) Example of a makefile (Net Express: MHP)

This subsection shows an example of a makefile needed for creating an MHP.

```
!include <ntwin32.mak>
# makefile for making UAP(COBOL Language)


# Command
STBMAKE          =      $(DCDIR)\bin\stbmake.exe

# User server object
MHP_SRC          =      CBLMHPM.obj TCPMHPSV.obj

# Include file
INCDIR           =       -I.                 \
                         -I $(DCDIR)\include

# Common object
SH_OBJ           =       $(DCDIR)\spool\trnrmcmd\userobj\mcf_sw.obj

# Common library
LIBNAME          =       $(DCDIR)\lib\libmcf.lib \
                         $(DCDIR)\lib\libmnet.lib \
                         $(DCDIR)\lib\libbetran.lib

# Library
CBLLIBNAME       =       cblrtss.lib cbllds.obj adis.obj adisinit.obj\
                         adiskey.obj mffh.obj

# Compiler options
CC               =       cl
COBOL            =       cobol
CCOPT            =       -nologo $(cflags) $(cvarsdll)
CONFIG           =       -c -Zl
CFLAGS           =       $(CCOPT) $(CONFIG) $(INCDIR)
CBLFG            =       /LITLINK /OBJ

# Linker options
LK               =       link
LNKFLG           =       $(conflags) /NODEFAULTLIB
LINKAGE          =      $(LIBNAME) msvcrt.lib user32.lib $(conlibsdll) $(CBLLIBNAME)
```

```
all : $(SH_OBJ) tcpmhp.exe

# Compile
.c.obj   :
                $(CC) $(CFLAGS) $*.c

TCPMHPSV.obj :
                $(COBOL) tcpmhpsv.cbl $(CBLFG);

CBLMHPM.obj :
                $(COBOL) cblmhpm.cbl $(CBLFG);

cblmhp_sstb.obj :
                $(CC) $(CFLAGS) cblmhp_sstb.c

# Create stub source file
cblmhp_sstb.c :
                $(STBMAKE) cblmhp.def

# Link
tcpmhp.exe : $(MHP_SRC) cblmhp_sstb.obj
                cblnames -V -MCBLMHPM + CBLMHPM
                $(LK) $(LNKFLG) $** $(LINKAGE) $(SH_OBJ) -out:$@
                !if exist $@.manifest mt.exe -nologo -manifest $@.manifest
-outputresource:$@;1
                !if exist $@.manifest del $@.manifest

$(SH_OBJ) :
                $(DCDIR)\bin\trnmkobj -o mcf_sw -R OpenTP1_MCF -C "/Zl"
```

3.3 Programming notes specific to OpenTP1 for Windows

This section presents information specific to OpenTP1 for Windows that we recommend you note before you begin programming.

(1) Notes on stub creation

For the stub source file name that is specified in the `stbmake` command, make sure that you specify the suffix `.def` that indicates an RPC interface definition file.

Use the Shift JIS character encoding for the RPC interface definition file.

(2) Notes about UAP execution in a single-thread environment

When an OpenTP1 UAP is executed in a single-thread environment, it must not create threads other than the main thread. If the UAP creates a thread other than the main thread (multi-thread environment), it might malfunction and terminate abnormally. Note also that, because UAPs under OpenTP1 run in a multi-thread environment through use of a DLL, you must specify `msvcrt.lib` as a linked runtime library when you create a UAP.

If you use code such as JNI (Java Native Interface) that creates threads, program operation cannot be guaranteed.

(3) Notes about UAP termination

Do not issue a function or method that terminates a process or a thread within a service function (SPP) or service method (SPP.NET). Issuing such a function or method might result in the following:

- Service functions (SPP) and service methods (SPP.NET) do not run as designed.
- Service requests to service groups fail.
- The UAP cannot be terminated.
- OpenTP1 cannot be terminated normally.

(4) Notes about using Visual Studio

- If you use a library created by using Visual Studio on the user server, make sure that you embed a manifest file in the library. If a manifest file is not embedded, the user server might be shut down with termination status `0xC0000135`.
- An object file created in a Visual Studio debug build might not run in an environment where Visual Studio is not installed. If you debug an object file in an environment where Visual Studio is not installed, check the contents of the linked DLL files and manifest file and create an appropriate manifest file. For details about manifest files, see the description of manifest files in the Visual Studio documentation.

(5) Notes about an environment that uses both Visual Studio and COBOL

In an environment in which Visual Studio and COBOL are both used, make sure that the environment variables for the language used for compilation are specified before the environment variables for any other language. If the environment variables for another language are specified first, unexpected linkage might result.

(6) Notes about using Net Express

- Of the arguments of OpenTP1's COBOL UAP creation program, numeric data must be declared with `COMP-5`, not `COMP`.
- When you execute a UAP, do not set a large value for the OpenTP1 console's buffer size and window size (height). If you specify a large value, the correct console display might not result and the UAP might not run successfully.
- If you run a UAP created with Net Express, OpenTP1 console's title characters might be changed.
- If you use a UAP created with Net Express to display messages on the OpenTP1 console by using the `DISPLAY` statement of the COBOL UAP creation program, and you log off the system during UAP execution, the UAP might be shut down. To display messages, use `CBLDCLOG('PRINT  ')` provided by OpenTP1.

3.4 Sample programs

This section explains the sample programs for OpenTP1 for Windows in terms of the differences from OpenTP1 for UNIX for each development language. For details about the sample programs themselves, see the OpenTP1 series manuals.

(1) In C language

The C language sample programs have the following differences from those for OpenTP1 for UNIX:

- The names of user service definition files for individual sample programs that are located in `%DCDIR%\examples\base\conf` are `bsespp_c` (SPP) and `bsesup_c` (SUP). Specify these file names in the argument of the `dcsvstart` command when you start the sample programs.
- OpenTP1 for Windows does not support the commands `chconf` (definition file change command) and `bkconf` (command for undoing the change made by `chconf`) that are provided in OpenTP1 for UNIX.
- If you use Visual Studio to create sample programs, the following message might be displayed:

  ```
  LINK : warning LNK4098: defaultlib 'LIBCMT' conflicts with
  another library. Use /NODEFAULTLIB:library.
  ```

  If this message is displayed, use the `/NODEFAULTLIB` option and make sure that only the `msvcrt.lib` runtime library is linked. If multiple types of runtime libraries are linked, the resulting program might not function as designed.

(2) In COBOL

The COBOL sample programs have the following differences from those for OpenTP1 for UNIX:

- The names of user service definition files for sample programs that are located in `%DCDIR%\examples\base\conf` are `bsespp_b` (SPP) and `bsesup_b` (SUP). Specify these file names in the argument of the `dcsvstart` command when you start the sample programs.
- OpenTP1 for Windows does not support the commands `chconf` (definition file change command) and `bkconf` (command for undoing the change made by `chconf`) that are provided in OpenTP1 for UNIX.
- When you create a UAP, execute the following command:
  - If you use COBOL2002, enter `nmake`.
  - If you use Net Express, enter `nmake -f make_mf`. The `link` command of Net Express might issue a `warning LNK4044` message, but this has no effect

on the operation.

- In the COBOL sample programs, `COMP-5`, not `COMP`, is used as the numeric data format of arguments. If you use `COMP-5` in COBOL2002, you must specify the `-Comp5` option during compilation. If you use Net Express and use a numeric data format other than `COMP-5`, the resulting program will not function correctly.

3.5 Handling of UAP shared libraries when using dynamic loading of service functions

This section discusses the handling of UAP shared libraries when dynamic loading of service functions is used. For an overview of dynamic loading of service functions, see the OpenTP1 series manuals.

Dynamic loading of service functions cannot be used in the following cases:

- SPP.NET server
- UAP shared library built by managed code

  Dynamic loading of service functions cannot call a service function contained in a UAP shared library built by managed code.

(1) Exporting UAP shared libraries

If service functions are in a UAP shared library, make sure that you export from the UAP shared library the service functions defined in the user service definition that are included in the UAP shared library when the user server starts.

If the service functions have not been exported (during process startup for a nonresident server), the following occurs:

For SPP

The `KFCA00344-E` message is output and then the `dc_rpc_mainloop` function returns with error `DCRPCER_FATAL(-303)`.

For MHP

The `KFCA00344-E` message is output and then the `dc_mcf_mainloop` function returns with error `DCMCFER_FATAL(-11902)`.

You can export any function other than the service functions specified in the user service definition according to the processing mode because export of such a function is not mandatory.

You can use the `lib` and `link` commands to export functions from a UAP shared library. Use one of the following methods to export service functions:

- Use `__declspec(dllexport)` to define the service function in the object file specified in the `lib` or `link` command name.
- Use the `lib` command to specify `/EXPORT:`*service-function-name*.
- Use the `lib` command to specify `/DEF:`*module-definition-file* and then define the service function in the `EXPORTS` statement format in the module definition file.

For examples of creation, see the following sample source (makefiles):

SPP development language and environment		Makefile name
C language		`%DCDIR%\examples\base\aplib\c\make_svdl`
COBOL	COBOL2002	`%DCDIR%\examples\base\aplib\cobol\make_svdl`
	Net Express	`%DCDIR%\examples\base\aplib\cobol\makemf_svdl`

(2) Handling of the library search path

If only a UAP shared library name is specified, OpenTP1 searches for the specified UAP shared library based on the OS library search path. In Windows, the specified UAP shared library is searched for using the search paths listed below (where the search paths are listed in order of priority):

- Folder containing the UAP
- UAP's current folder (`%DCDIR%\tmp\home\`*each-server's-folder*)#1
- Windows system folder#2
- Windows folder#2
- Folder specified in the `prcsvpath` definition command in the process service definition

  If multiple folders are specified, OpenTP1 uses the UAP shared library found in the first folder searched.

#1

This is a temporary folder managed by OpenTP1. Do not store a created UAP shared library in this folder because this folder is deleted when the OpenTP1 system or UAP starts.

#2

We strongly recommend that you do not store a created UAP shared library in a Windows system folder or a Windows folder, so make sure that you avoid storing created UAP shared libraries in these folders.

The following example loads the into the UAP the UAP shared library located under folder (b):

Name of path containing UAP: `c:\user\aplib\user.exe`
Name of path containing DLL: `c:\opentp1\aplib\libusr.dll`
`c:\user\aplib\libusr.dll`
<Process service definition>
`prcsvpath c:\opentp1\aplib; c:\user\aplib`
(a) (b)

(3) Replacing the service functions of user service processes

When you are using dynamic loading of service functions, you can replace a user server's service functions without having to stop online operation. This subsection explains how to replace service functions and how to check the result.

(a) Replacing service functions

This subsection explains how to replace service functions when you change the user service definition and when you change the UAP shared library search path.

When you change the user service definition:

1. Use the `dcsvstop` command to terminate the applicable user server.
2. Change the `service` operand specification in the user service definition for the user server.
3. Use the `dcsvstart` command to start the user server.

When you change the UAP shared library search path:

1. Use the `dcsvstop` command to terminate the applicable user server.
2. Place a UAP shared library in a new UAP shared library storage folder.
3. Use the `prcpath` command to change the UAP shared library search path to the new UAP shared library storage folder.
4. Use the `dcsvstart` command to start the user server.

(b) Checking the result after replacing a service function

Check the start time of the new executable file and compare it to the execution time of the `dcsvstart` or `scdrsprc` command.

You can use the procedure described below to check the start time of the new executable file. The example provided here is for Windows Server 2008.

To check the result:

1. From the **Start** menu, choose **Run**.
2. In **Open**, enter `Msinfo32.exe` and then click **OK**. The System Information dialog box opens.
3. In the left-hand pane of the System Information dialog box, click **Software Environment**, and then **Running Tasks**.
4. From the files displayed in the right-hand pane, find the executable file and check its start time.

(4) Specifying the UAP shared library name

You can specify both an absolute path and a library name in the `service` operand in

the user service definition.

If the UAP shared library specified by the absolute path is the same as that specified by the library name, the UAP shared library path to be loaded depends on the specification order.

The following provides the details based on which name is specified first.

If the first UAP shared library name specified is a library name:

The UAP shared libraries with the specified library name are searched for in the library path according to the search order of paths specified for the library name. As a result, the UAP shared library under the first folder found is loaded.

Next, the UAP shared library specified by the absolute path is loaded.

In this specification order, two different UAP shared libraries are loaded into the UAP.

If the UAP shared library storage folder specified by the absolute path is searched first, the same UAP shared library is loaded twice.

If the first UAP shared library name specified is an absolute path:

At first, the UAP shared library specified by the absolute path is loaded. Next, an attempt is made to load the UAP shared library specified as the library. However, the UAP shared library specified by the absolute path is loaded again because the UAP shared library with the same name has already been loaded by using the absolute path specification. In this specification order, the UAP shared library specified by the absolute path is loaded twice into the UAP.

(5) Notes about UAP shared libraries when using dynamic loading of service functions

Do not use a thread local storage (TLS) to declare variables in a UAP shared library specified in the `service` operand in the user service definition. If the UAP shared library contains a UAP object that contains the code shown below, referencing of the corresponding variable results in an access exception:

```
__declspec(Thread) int sample;
```

Chapter

4. Environment Setup

The environment setup procedure differs between OpenTP1 for Windows and OpenTP1 for UNIX. This chapter explains the environment settings specific to OpenTP1 for Windows. For the details common to OpenTP1 for Windows and OpenTP1 for UNIX, see the OpenTP1 series manuals.

4.1 Environment setup procedure
4.2 Registering the OpenTP1 administrator
4.3 Creating OpenTP1 system definition information
4.4 Environment variable settings
4.5 Configuring an OpenTP1 file system
4.6 Settings for linking databases
4.7 OS settings
4.8 MultiOpenTP1 settings

4.1 Environment setup procedure

This section explains the environment setup procedure up to the point immediately before OpenTP1 startup. The following figure shows the environment setup procedure for OpenTP1 for Windows.

Figure 4-1: Environment setup procedure for OpenTP1 for Windows

Environment setup procedure	Section in this manual	OpenTP1 series manual
Registering the OpenTP1 administrator	4.2	*OpenTP1 Operation*
Installing OpenTP1	—#1	—#1
Creating system definitions	4.3	*OpenTP1 System Definition* *OpenTP1 Operation*
Specifying environment variables	4.4	—
Creating an OpenTP1 file system area	4.5	*OpenTP1 Description* *OpenTP1 Operation*
Initializing the OpenTP1 file system	4.5	*OpenTP1 Operation*
Creating OpenTP1 files	—	*OpenTP1 Description* *OpenTP1 Operation*
Creating non-OpenTP1 files	—	*OpenTP1 Description* *OpenTP1 Operation*
Registering the resource manager	4.6#2	*OpenTP1 Operation*
Creating a transaction control object file	4.6#2	*OpenTP1 Operation*
Specifying OS settings	4.7	—

Legend:

: Required step

: Optional step

— : Not applicable

#1: For details about the installation, see the product's Release Notes.
#2: These sections explain the settings required in order to link OpenTP1 for Windows to each database. For details about the settings for other resource managers, see the manual *OpenTP1 Operation*.

The OpenTP1 administrator sets up an environment for OpenTP1 for Windows. For the details of setting up an environment, see the appropriate sections in this chapter and the OpenTP1 series manuals, as indicated in the figure.

For details about the settings for using multiOpenTP1, see *4.8 MultiOpenTP1 settings*.

Note:

If you use functions provided by other products, such as TP1/Message Control or TP1/Message Queue, you must perform the procedures specific to each product in addition to the procedures indicated in the figure. These specific procedures are common to OpenTP1 for Windows and OpenTP1 for UNIX. For details, see the overview of the environment setup procedure in the manual *OpenTP1 Version 7 Operation*.

Reference note:

You can use a graphical user interface (GUI) to perform the minimum environment setup for OpenTP1 for Windows (such as creating system definitions) that is needed in order to use OpenTP1. For details about the OpenTP1 environment GUI setup procedure, see *6.7 GUI provided by OpenTP1 for Windows*.

4.2 Registering the OpenTP1 administrator

In OpenTP1 for Windows, the OpenTP1 administrator must be a user with Administrators group permissions. There are no restrictions on the name assigned to the OpenTP1 administrator.

All other information is common to OpenTP1 for Windows and OpenTP1 for UNIX. For details, see the manual *OpenTP1 Version 7 Operation*.

4.3 Creating OpenTP1 system definition information

This section explains the differences between creating the OpenTP1 system definition in OpenTP1 for Windows, and creating the information in OpenTP1 for UNIX. For details common to OpenTP1 for Windows and OpenTP1 for UNIX, see the manuals *OpenTP1 Version 7 System Definition* and *OpenTP1 Version 7 Operation*.

4.3.1 Creating definition files

You use a text editor to create the OpenTP1 system definition files. Note the following about creating the definition files:

- Use the Shift JIS encoding for the definition files.
- Make sure that each line in the definition files ends with an end-of-line code. Some text editors do not automatically insert an EOF control character at the end of a file. In such a case, the definition specified on the last line of the definition file might not be recognized.

You can use the `dcdefchk` command to check the created system definitions.

For details about the system definitions and checking the definitions, see the manual *OpenTP1 Version 7 System Definition*.

4.3.2 Changing definitions

The method for changing system definitions differs between OpenTP1 for Windows and OpenTP1 for UNIX. Use the method described below in OpenTP1 for Windows.

To change system definitions:

1. Terminate OpenTP1 normally.
2. Make changes to the system definitions.
3. Start OpenTP1 normally.

The other details are the same in OpenTP1 for Windows and OpenTP1 for UNIX. For details about how to change definitions, see the manual *OpenTP1 Version 7 System Definition*.

4.3.3 Setting the runtime library path

If you use runtime libraries other than those provided by OpenTP1, Windows, and Microsoft Visual Studio in order to use COBOL UAPs or to link databases, you must set the paths of those runtime libraries.

(1) Setting the path

Use the `prcsvpath` process service definition command or the `prcpath` command to specify the runtime library storage folder.

For details about the `prcsvpath` process service definition command, see Chapter *5. System Definitions*.

For details about the `prcpath` command, see Chapter *7. Operation Commands*.

(2) Response when the user server shuts down immediately after startup

If the user server shuts down with termination status `0x00000080` or `0xC0000135` immediately after startup, the path for the runtime library that was to be loaded by the user server during startup might not have been set correctly. Check the runtime library that was linked when the user server was created and specify in the `prcsvpath` process service definition command the path of the required runtime library storage folder.

If you open the user server storage folder in Explorer and double-click the EXE file that was shut down, an error dialog box might be displayed in which is shown the name of the runtime library that failed to load. Note, however, that if the runtime library storage folder is specified in the `PATH` environment variable, the runtime library name will not be displayed. In such a case, you can use the `dumpbin` command of Visual Studio to determine the name of the runtime library loaded by the EXE file. The following shows an example of using the `dumpbin` command to check runtime libraries:

```
dumpbin /DEPENDENTS %DCDIR%\aplib\basespp.exe
```

4.4 Environment variable settings

In OpenTP1 for Windows, you must specify environment variables in the login environment in order to execute OpenTP1 commands. The following table lists and describes the environment variables for OpenTP1 for Windows.

Table 4-1: Environment variables for OpenTP1 for Windows

Environment variable name	Setting
DCCONFPATH#	Specifies (in 1 to 246 bytes) the absolute path name of the folder containing the system definition files. A path name cannot contain any spaces. Within a machine, all DCCONFPATH environment variables must be the same. A path name is case sensitive.
DCDIR	There is no need for the user to set the DCDIR environment variable. When you set up OpenTP1, the path name of the OpenTP1 folder is set automatically in the DCDIR environment variable. Do not set the DCDIR environment variable in a definition file or a batch file. If you need to set this environment variable for system operation purposes, note that the value is case-sensitive and that the character string that is set must be exactly the same as that for the DCDIR environment variable.
DCUAPCONFPATH#	Specifies (in 1 to 246 bytes) the absolute path name of the folder for storing the user service definition files when you wish to store these files in a different folder from the one specified in the DCCONFPATH environment variable. A path name cannot contain any spaces.
INCLUDE	Specifies %DCDIR%\include if user server programs are to be created.
LIB	Specifies %DCDIR%\lib if the trnlnkrm command is to be executed or user server programs are to be created.
PATH	When OpenTP1 is set up, OpenTP1's executable file storage path (%DCDIR%\bin) is added automatically to the PATH environment variable. Do not delete the %DCDIR%\bin specification. If it is deleted, OpenTP1 will not function correctly. If the PATH environment variable has already been set, %DCDIR%\bin will be added at the end. If the path of a drive connected to the network is specified before %DCDIR%\bin, OpenTP1 will not start because of an application error. Therefore, specify the path of a drive connected to the network after %DCDIR%\bin.
TZ	Specifies the time zone. Make sure that the value of this environment variable matches the time zone value in **Time zone** in **Control Panel**. For example, if the time zone is (GMT+9:00) Osaka, Sapporo, Tokyo, (GMT+9:00) Seoul, or (GMT+9:00) Yakutsk, the value of the TZ environment variable must be set to JST-9. If the TZ environment variable is not set or no time is set within the system, the correct time might not be obtained internally in OpenTP1.

#

When the system starts, OpenTP1 analyzes the files in the folder specified in the DCCONFPATH or DCUAPCONFPATH environment variable as definition files. If the specified folder contains files that are not definition files used by OpenTP1, it might take a long time for OpenTP1 to start. Therefore, do not store any files that are not definition files used by OpenTP1 in the folder specified in the DCCONFPATH and DCUAPCONFPATH environment variables.

4.5 Configuring an OpenTP1 file system

This section explains the configuration details of an OpenTP1 file system in OpenTP1 for Windows that differ from OpenTP1 for UNIX. For the details common to OpenTP1 for Windows and OpenTP1 for UNIX, see the manuals *OpenTP1 Version 7 Description* and *OpenTP1 Version 7 Operation*.

4.5.1 OpenTP1 file system

An OpenTP1 file system is a file system dedicated to OpenTP1, independent from the file systems provided by the OS. In OpenTP1 for Windows, an OpenTP1 file system is configured either in regular Windows files or in an area to which the Windows direct disk access (raw I/O) function (raw I/O facility) is applied.

This subsection explains the characteristics of an OpenTP1 file system and how to choose an OpenTP1 file system. It also provides notes about an OpenTP1 file system in OpenTP1 for Windows. You choose regular Windows files or an area to which the raw I/O facility is applied to configure the OpenTP1 file system by taking into account the characteristics of the OpenTP1 file system.

For details about configuring an OpenTP1 file system by using regular Windows files, see the manuals *OpenTP1 Version 7 Description* and *OpenTP1 Version 7 Operation*. For details about configuring an OpenTP1 file system by applying the raw I/O facility, see *4.5.2 Configuring the OpenTP1 file system by applying the raw I/O facility*.

(1) Characteristics of an OpenTP1 file system

This subsection explains write and read performance on an OpenTP1 file system during online OpenTP1 operation.

- Write performance on an OpenTP1 file system#
  - When regular Windows files are used

    Data is flushed to disk each time a write operation occurs in order to prevent data loss during Windows delayed write processing. Performance cost is higher when disk flush is performed than when only the write operation is performed.
  - When the raw I/O facility is applied

    Performance cost is lower than when regular Windows files are used because flush processing is not needed each time a write operation occurs.

#

Disk flush processing is not performed when data is written to a file other than an OpenTP1 file system (such as a `dclog` file).

- Read performance on the OpenTP1 file system
  - When regular Windows files are used

    Read processing is performed via the Windows system cache.
  - When the raw I/O facility is applied

    Each time a read operation occurs, data is read from the disk.

  Read performance is better using regular Windows files than an area to which the raw I/O facility is applied.

(2) Notes about choosing an OpenTP1 file system

Note the following about choosing an OpenTP1 file system:

- Unload and backup processing

  Because frequent read operations are involved in loading journal files and backing up OpenTP1 files, the command execution time required to process journal files and to back up files can be reduced to a greater extent when regular Windows files are used than when the raw I/O facility is applied.

  When unloading and backup performance is important, we recommend that you configure the OpenTP1 file system in regular Windows files.

- I/O processing during online OpenTP1 operation

  On the following files, write processing occurs more frequently than read processing during online OpenTP1 operation:

  - System journal files
  - Checkpoint dump files
  - Status files
  - Server recovery journal files
  - XAR files

  If online performance is important, we recommend that you configure the OpenTP1 file system in an area to which the raw I/O facility is applied.

- OpenTP1 restart processing

  Because OpenTP1 restart processing involves considerable data read processing (such as from journal files), you might be able to reduce the restart processing time by using regular Windows files.

- Processing in cache blocks

  Performance of the following processing is better when regular Windows files are used than when the raw I/O facility is applied:

- Reading blocks that are not in cache blocks
- Processing that involves frequent reuse and release of cache blocks

Performance of the following processing is the same whether regular Windows files are used or the raw I/O facility is applied:

- Reading blocks that are in cache blocks

■ I/O processing by the DAM service

The performance of processing that issues the `dc_dam_rewrite` function frequently is better when the raw I/O facility is applied than when regular Windows files are used.

If `flush` is specified in the `dam_update_block_over` operand in the DAM service definition, the `dc_dam_write` function processing involves read processing. If the block specified in the `dc_dam_write` function is in a cache block, disk read processing does not occur.

If processing involves frequent reuse and release of cache blocks, write performance is better when the raw I/O facility is applied. Therefore, the performance of `dc_dam_write` function processing is better when the raw I/O facility is applied than when regular Windows files are used. However, if `flush` is not specified in the `dam_update_block_over` operand, performance is better when regular Windows files are used.

If performance of the `dc_dam_write` and `dc_dam_rewrite` functions is important, we recommend that you configure DAM files in an area to which the raw I/O facility is applied. If performance of the `dc_dam_read` function is important, we recommend that you configure the DAM files in regular Windows files.

■ I/O processing by the TAM service

The TAM service performs I/O processing mainly on TAM tables in shared memory. Real I/O processing occurs in TAM files in the following cases:

- Loading TAM tables
- Unloading TAM tables
- Real update processing at specific intervals
- Real update processing when a checkpoint dump is acquired

In any of these cases, I/O processing is performed asynchronously with user server processing. The user server's TAM access will never wait for I/O processing. Therefore, performance of the `dc_tam_read`, `dc_tam_rewrite`, and `dc_tam_write` functions is the same whether regular Windows files are used or the raw I/O facility is applied.

Note:

The difference in performance between using regular Windows files and when the raw I/O facility is applied also depends on the disk device being used. For some disk devices, there is almost no difference in performance.

(3) Notes on the OpenTP1 file system

Note the following points concerning the OpenTP1 file system in OpenTP1 for Windows:

- Location of OpenTP1 file system

  Do not configure the OpenTP1 file system on a network drive. In order to guarantee disk write processing, configure the OpenTP1 file system on a local disk or a shared disk.

- Protection of OpenTP1 files

  Protection of OpenTP1 files is not supported in OpenTP1 for Windows.

4.5.2 Configuring the OpenTP1 file system by applying the raw I/O facility

This subsection explains how to configure the OpenTP1 file system by applying the raw I/O facility.

In OpenTP1 for Windows, you can create an OpenTP1 file system area by applying Windows direct disk access (raw I/O). This function is called the *raw I/O facility*. The raw I/O facility enables you to access partitions and logical drives in the same manner as with files.

You can use the raw I/O facility for the following purposes:

- Configuring the OpenTP1 file system
- Restoring the OpenTP1 file system

(1) How to configure the OpenTP1 file system

To configure the OpenTP1 file system in an area to which the raw I/O facility is applied:

1. Prepare an unformatted partition.

   Use Windows **Disk Management** to create a partition. For details about how to create partitions, see the Windows **Disk Management** Help.

2. Execute the `filmkfs` command to initialize the prepared partition or logical drive so that it can be used as the OpenTP1 file system.

   The following shows an example of the `filmkfs` command:

```
filmkfs -s 512 -n 7 -l 100 X:
```

This example uses drive X as the partition to which the raw I/O facility is applied (raw I/O partition).

For details about the filmkfs command, see Chapter *7. Operation Commands*.

The following notes apply to configuring the OpenTP1 file system in an area to which the raw I/O facility is applied.

Notes about preparing the partition

- Assign a drive letter to the partition.
- The raw I/O facility can be applied only to a fixed disk drive whose sector length is 512 bytes.

Notes about executing the filmkfs command

- When you apply the raw I/O facility, make sure that you specify the -s option in the filmkfs command. Specify 512 as the sector length in the -s option.
- In the filmkfs command, specify the path name for the partition or logical drive in the format *drive-letter*:. If the specified path name is for a partition that has already been formatted, the filmkfs command issues the KFCA01537-E message and returns an error.

(2) Specifying OpenTP1 files

The OpenTP1 file system area allocated as a raw I/O partition is displayed in the format *drive-letter*:. When you specify an OpenTP1 file, use its absolute path in the same manner as for regular Windows files.

Examples of OpenTP1 file specifications are shown below. In these examples, drive X is allocated to the raw I/O partition.

- Example of system journal file creation

```
jnlinit -j jnl -f X:\jnlf01 -n 1024
```

- Example of system journal file specification

```
jnladdpf -g jnlgrp01 -a X:\jnlf01
```

(3) Deleting the OpenTP1 file system

You use Windows **Disk Management** to completely delete an OpenTP1 file system that has been configured as a raw I/O partition. For details about how to delete partitions, see the Windows **Disk Management** Help.

(4) Setting a shared disk when the system switchover facility is used

In Windows Server 2003, an OpenTP1 file system configured as a raw I/O partition

can be used by the system switchover facility. Note that raw I/O partitions cannot be used by the system switchover facility in Windows Server 2008 or later.

In MSCS, raw I/O partitions cannot be specified as shared disk resources. Therefore, you must use the following procedure to set shared disk resources in such a case.

To set shared disk resources:

1. Create a new NTFS partition on the same physical disk as the raw I/O partition.

   There is no need to configure the OpenTP1 file system in the created NTFS partition.

2. Specify the created NTFS partition as a shared disk resource in the OpenTP1 resource group.

3. Set the shared disk resource in the general-purpose service OpenTP1 dependencies.

4.6 Settings for linking databases

OpenTP1 for Windows can link to the following databases:

- HiRDB
- Oracle
- SQL Server

This section explains the settings required for linking each database in OpenTP1 for Windows.

4.6.1 Linking with HiRDB

This subsection explains the settings required to link with HiRDB.

(1) Registering HiRDB

You must register HiRDB in the OpenTP1 system. Use the `trnlnkrm` command to do this.

The following shows an example of the `trnlnkrm` command for HiRDB Version 8:

```
trnlnkrm -a HiRDB_DB_SERVER -s pdtxa_switch -o pdcltx32.lib
```

For details about the `trnlnkrm` command, see Chapter *7. Operation Commands*. For details about the information to be specified in the `trnlnkrm` command, such as RM name, RM switch name, and RM-related object name, see the manual *Scalable Database Server HiRDB Version 7 Installation and Design Guide*, for Windows systems.

The following notes explain execution of the `trnlnkrm` command.

- `-o` option in the `trnlnkrm` command

  Set the path of a related object in the `LIB` environment variable beforehand. If you specify the absolute path, including folders, for the related object, use `\\` as the folder delimiter.

- Specification of the `prcsvpath` definition command

  When you register HiRDB in the OpenTP1 system, the transaction service process uses HiRDB runtime libraries. Therefore, you must specify the folder containing the HiRDB runtime libraries in the `prcsvpath` process service definition command. The following shows an example of the `prcsvpath` definition command:

```
prcsvpath c:\hirdb_s\client\utl
```

(2) Specifying environment variables

You must set appropriate values in HiRDB environment variables in order to access HiRDB. Specify the environment variables in `putenv` format in the following OpenTP1 system definitions:

- Transaction service definition
- User service definition
- User service default definition

The following shows an example of a transaction service definition:

```
putenv PDHOST host1
putenv PDNAMEPORT 22200
putenv PDTMID smpl
putenv PDSWAITTIME 6000
```

The following shows an example of a user service definition or user service default definition:

```
putenv PDHOST host1
putenv PDNAMEPORT 22200
putenv PDTMID smpl
putenv PDSWAITTIME 6000
putenv PDUSER '"root"/"root"'
putenv PDXAMODE 1
```

Note:

Many of the errors that occur when a database is accessed from OpenTP1 while using the X/Open XA interface are due to invalid settings of environment variables (such as `PDHOST` and `PDNAMEPORT`). If the `KFCA00901-W` message is issued, check the settings of environment variables and revise them as necessary.

For details about the OpenTP1 system definition, see Chapter *5. System Definitions*. For details about the environment variables to be specified and their settings, see the settings used for installing HiRDB and the manual *Scalable Database Server HiRDB Version 7 UAP Development Guide*, for Windows systems.

(3) Specifying the transaction service definition

Use the `trnstring` transaction service definition command to specify the information required in order to access the resource manager. When you link OpenTP1 with HiRDB, specify the `-n` option and, if necessary, the `-d` option in the `trnstring` definition command.

There is no need to specify the `xa_open` and `xa_close` function strings (specified in the `-o`, `-O`, `-c`, and `-C` options in the `trnstring` definition command).

The following shows an example of a transaction service definition:

```
trnstring -n HiRDB_DB_SERVER
```

(4) Compiling and linking UAPs

When you compile and link a UAP that accesses HiRDB, you must use the `trnmkobj` command to link the transaction control object created under `%DCDIR%\spool\trnrmcmd\userobj`.

For details about how to compile and link UAPs, see *3.2 Compiling and linking UAPs*, the edition of the manual *OpenTP1 Programming Reference* for the applicable language, and the *Scalable Database Server HiRDB Version 7 UAP Development Guide*, for Windows systems. For details about the `trnmkobj` command, see Chapter *7. Operation Commands*.

The following shows examples of creating a transaction control object for UAPs that access HiRDB in C language and in COBOL.

In C language

```
trnmkobj -o rm_obj -r HiRDB_DB_SERVER
```

In COBOL

```
trnmkobj -o rm_obj -C "/Zl" -r HiRDB_DB_SERVER
```

(5) Example of a makefile (in C language)

This subsection presents an example of a makefile that is used when an SPP is created in C language.

(a) Source file names

The source file names used in the example makefile are as follows:

- Source program names of UAP created in C language:
  - `exmain.c` (main function)
  - `exsv1.c` (service function 1)
  - `exsv2.c` (service function 2)
  - `exproc.ec` (`ec` file)
- RPC interface definition file name: `ex.def`

(b) Example of a makefile

The following shows an example of a makefile:

```
NODEBUG = 1
!include <ntwin32.mak>
#TP1TEST Make File
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
HIRDBLIB = pdcltx32.lib

all:TP1TEST.exe
# Compile
exmain.obj:exmain.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsv1.obj:exsv1.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsv2.obj:exsv2.c
        $(cc) $(cflags) $(cvarsdll) $*.c
ex_sstb.obj:ex_sstb.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exproc.obj:exproc.c
        $(cc) $(cflags) $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
        $(DCDIR)\bin\stbmake $?
# Precompile
exproc.c:exproc.ec
        $(PDDIR)\client\utl\pdcpp /XA $?
# Link
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj exproc.obj
        $(link) $(conflags) -out:$*.exe $** \
        [$(DCDIR)\spool\trnrmcmd\userobj\rm_obj.obj \]#3
        $(TP1LIB) $(SYSLIB) $(HIRDBLIB)
```

#1

Specify `libdam.lib` only if you use TP1/FS/Direct Access.

#2

Specify `libtam.lib` only if you use TP1/FS/Table Access.

#3

If you use TP1/FS/Direct Access or TP1/FS/Table Access, add `OpenTP1_DAM` or `OpenTP1_TAM` when you use the `trnmkobj` command to create a transaction control object.

(6) Example of a makefile (for COBOL)

This subsection presents an example of a makefile that is used when an SPP is created in COBOL.

(a) Source file names

The source file names used in the example makefile are as follows:

- Source program name of UAP created in COBOL2002: `main.cbl`

- Service program name of UAP created in COBOL2002: `sev1.cbl`
- Service program name of UAP created in COBOL2002: `sev2.cbl`
- `ecb` file name 1: `exprocb1.ecb`
- `ecb` file name 2: `exprocb2.ecb`
- RPC interface definition file name: `ex.def`

(b) Example of a makefile

The following shows an example of a makefile:

```
!include <ntwin32.mak>
#TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
HIRDBLIB = pdcltx32.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
# Compile
main.obj:main.cbl
        $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
        $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
        $(CCBL) $(CBLFG2) sev2.cbl
exprocb1.obj:exprocb1.cbl
        $(CCBL) $(CBLFG2) exprocb1.cbl
exprocb2.obj:exprocb2.cbl
        $(CCBL) $(CBLFG2) exprocb2.cbl
ex_sstb.obj:ex_sstb.c
        $(cc) $(cflags) /Zl $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
        $(DCDIR)\bin\stbmake $?
# Precompile
exprocb1.cbl:exprocb1.ecb
        $(PDDIR)\client\utl\pdcbl /XA $?
exprocb2.cbl:exprocb2.ecb
        $(PDDIR)\client\utl\pdcbl /XA $?
# Link
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj \
             exprocb1.obj exprocb2.obj
        $(LK) -OutputFile $*.exe $** \
        [$(DCDIR)\spool\trnrmcmd\userobj\rm_obj.obj \]#3
        $(TP1LIB) $(SYSLIB) $(HIRDBLIB)
```

#1

Specify `libdam.lib` only if you use TP1/FS/Direct Access.

#2

Specify `libtam.lib` only if you use TP1/FS/Table Access.

#3

If you use TP1/FS/Direct Access or TP1/FS/Table Access, add `OpenTP1_DAM` or `OpenTP1_TAM` when you use the `trnmkobj` command to create a transaction control object.

(7) Notes about linking HiRDB

- If you link OpenTP1 and HiRDB while using the X/Open XA interface, make sure that all HiRDB-related services are running.
- If you link OpenTP1 and HiRDB while using the X/Open XA interface, use the following OpenTP1 functions to start transactions and acquire synchronization points:

  `dc_trn_begin`, `tx_begin`, `dc_trn_unchained_commit`, `dc_trn_chained_commit`, `tx_commit`, `dc_trn_unchained_rollback`, `dc_trn_chained_rollback`, `tx_rollback`

  If HiRDB's SQL language is used to perform commit and rollback processing or if HiRDB linked with the XA interface makes an attempt to use an unsupported function, transactions might not be processed correctly, resulting in a data integrity error in resources or in shutdown of the OpenTP1 system.

4.6.2 Linking with Oracle

This subsection explains the settings required to link with Oracle.

(1) Registering Oracle

You must register Oracle in the OpenTP1 system. Use the `trnlnkrm` command to do this.

The following shows an example of the `trnlnkrm` command for Oracle Database 11g:

```
trnlnkrm -a Oracle_XA -s xaosw -o oraxa11.lib
```

For details about the `trnlnkrm` command, see Chapter *7. Operation Commands*. For details about the information to be specified in the `trnlnkrm` command, such as RM name, RM switch name, and RM-related object name, see the Oracle documentation.

The following notes explain execution of the `trnlnkrm` command.

- `-o` option in the `trnlnkrm` command

  Set the path of a related object in the `LIB` environment variable beforehand. If you specify the absolute path including folders for the related object, use `\\` as the folder delimiter.

- Specification of the prcsvpath definition command

  When you register Oracle in the OpenTP1 system, the transaction service process uses Oracle runtime libraries. Therefore, you must specify the folder containing the Oracle runtime libraries in the prcsvpath process service definition command. The following shows an example of the prcsvpath definition command:

```
prcsvpath c:\orant\bin
```

(2) Specifying environment variables

You must set appropriate values in the Oracle environment variables in order to access Oracle. Specify the environment variables in putenv format in the following OpenTP1 system definitions:

- Transaction service definition
- User service definition
- User service default definition

The following shows an example of these definitions:

```
putenv ORACLE_HOME c:\orant
putenv ORACLE_SID ORCL
```

For details about the OpenTP1 system definition, see Chapter *5. System Definitions*. For details about the environment variables to be specified and their settings, see the settings used for installing Oracle and the Oracle documentation.

(3) Specifying the transaction service definition

Use the trnstring transaction service definition command to specify the information required in order to access the resource manager. When you link OpenTP1 with Oracle, specify the -n, -o, and -O options and, if necessary, the -d option in the trnstring definition command.

Specify in the -o and -O options the xa_open function string used to access the resource manager. There is no need to specify the xa_close function string (specified in the -c and -C options in the trnstring definition command).

For uid in the xa_open function string for the transaction service that is specified in the -o option, specify the Oracle database administrator. If the specified uid does not belong to the Oracle database administrator or the specified uid does not have the DBA privilege, partial UAP recovery processing or transaction recovery processing after an OpenTP1 system restart might not be performed successfully. For details about the xa_open function string to be specified in Oracle, see the Oracle documentation.

The following shows an example of a transaction service definition:

```
trnstring -n Oracle_XA \
-o "Oracle_XA+Acc=P/sys/change_on_install+SesTm=60" \
-O "Oracle_XA+Acc=P/scott/tiger+SesTm=60"
```

Note

If a character string enclosed in double quotation marks (") continues onto the next line following a continuation line symbol (\), make sure that the specification on the continuation line begins in column 1.

Note:

Many of the errors that occur when an Oracle database is accessed from OpenTP1 while using the X/Open XA interface are due to an invalid `xa_open` function string definition. If the `KFCA00901-W` message is issued, check the `xa_open` function string definition and revise it as necessary.

(4) Settings required for accessing multiple Oracle databases

This subsection explains the settings required in order to access multiple Oracle databases from OpenTP1. These settings also apply when a single Oracle database is accessed using multiple Oracle `uids`.

- Setting up SQL*Net

  If OpenTP1 is to access multiple Oracle databases, SQL*Net is required according to the specifications for Oracle open and close strings. Set up SQL*Net by referencing the Oracle documentation.

- Defining the `xa_open` function string

  If OpenTP1 is to access multiple Oracle databases, the method for defining the `xa_open` function string used for accessing the resource manager varies as described below.

  1. Define the `-i` option in the `trnstring` transaction service definition command so that multiple Oracle databases can be identified by *resource-manager-name* + *resource-manager-extension*.
  2. In the `-i` option of the `trnrmid` user service default definition command, define the resource manager extension that was defined in step 1.

     The following shows an example of a transaction service definition:

```
trnstring -n Oracle_XA -i s1 \
-o "Oracle_XA+Acc=P/sys/change_on_install+SesTm=60" \
-O "Oracle_XA+Acc=P/scott/tiger+SesTm=60"
trnstring -n Oracle_XA -i s2 \
-o "Oracle_XA+SqlNet=host01+Acc=P/sys/change_on_install+SesTm=60"\
-O "Oracle_XA+SqlNet=host01+Acc=P/scott/tiger+SesTm=60"
```

The following shows an example of a user service default definition:

```
trnrmid -n Oracle_XA -i s1
```

The following shows an example of a user service definition:

```
trnrmid -n Oracle_XA -i s2
```

Note

The `trnrmid` definition command takes effect for each resource manager extension. If `s1` is specified in the user service default definition and `s2` is specified in the user service definition, both `s1` and `s2` take effect in the user service definition.

(5) Compiling and linking UAPs

When you compile and link a UAP that accesses Oracle, you must use the `trnmkobj` command to link the transaction control object created under `%DCDIR%\spool\trnrmcmd\userobj`.

For details about how to compile and link UAPs, see *3.2 Compiling and linking UAPs*, the edition of the manual *OpenTP1 Programming Reference* for the applicable language, and the Oracle documentation. For details about the `trnmkobj` command, see Chapter *7. Operation Commands*.

The following shows examples of creating a transaction control object for UAPs that access Oracle in C language and in COBOL.

In C language

```
trnmkobj -o rm_obj -r Oracle_XA
```

In COBOL

```
trnmkobj -o rm_obj -C "/Zl" -r Oracle_XA
```

(6) Example of a makefile (in C language)

This subsection presents an example of a makefile that is used when an SPP is created in C language.

(a) Source file names

The source file names used in the example makefile are as follows:

- Source program names of UAP created in C language:
  - `exmain.c` (main function)
  - `exsv1.c` (service function 1)

- exsv2.c (service function 2)
- exproc.pc (pc file)

■ RPC interface definition file name: ex.def

(b) Example of a makefile

The following shows an example of a makefile:

```
NODEBUG = 1
!include <ntwin32.mak>
#TP1TEST Make File
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
ORACLELIB = oraxa11.lib

all:TP1TEST.exe
# Compile
exmain.obj:exmain.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsv1.obj:exsv1.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsv2.obj:exsv2.c
        $(cc) $(cflags) $(cvarsdll) $*.c
ex_sstb.obj:ex_sstb.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exproc.obj:exproc.c
        $(cc) $(cflags) $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
        $(DCDIR)\bin\stbmake $?
# Precompile
exproc.c:exproc.pc
        $(ORACLE_HOME)\bin\proc release_cursor=yes $?
# Link
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj exproc.obj
        $(link) $(conflags) -out:$*.exe $** \
        [$(DCDIR)\spool\trnrmcmd\userobj\rm_obj.obj \]#3
        $(TP1LIB) $(SYSLIB) $(ORACLELIB)
```

#1

Specify libdam.lib only if you use TP1/FS/Direct Access.

#2

Specify libtam.lib only if you use TP1/FS/Table Access.

#3

If you use TP1/FS/Direct Access or TP1/FS/Table Access, add OpenTP1_DAM or OpenTP1_TAM when you use the trnmkobj command to create a transaction control object.

(7) Example of a makefile (for COBOL)

This subsection presents an example of a makefile that is used when an SPP is created in COBOL.

(a) Source file names

The source file names used in the example makefile are as follows:

- Source program name of UAP created in COBOL2002: `main.cbl`
- Service program name of UAP created in COBOL2002: `sev1.cbl`
- Service program name of UAP created in COBOL2002: `sev2.cbl`
- `pco` file name 1: `exprocb1.pco`
- `pco` file name 2: `exprocb2.pco`
- RPC interface definition file name: `ex.def`

(b) Example of a makefile

The following shows an example of a makefile:

```
!include <ntwin32.mak>
#TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
ORACLELIB = oraxa11.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
# Compile
main.obj:main.cbl
        $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
        $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
        $(CCBL) $(CBLFG2) sev2.cbl
exprocb1.obj:exprocb1.cbl
        $(CCBL) $(CBLFG2) exprocb1.cbl
exprocb2.obj:exprocb2.cbl
        $(CCBL) $(CBLFG2) exprocb2.cbl
ex_sstb.obj:ex_sstb.c
        $(cc) $(cflags) /Zl $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
        $(DCDIR)\bin\stbmake $?
# Precompile
exprocb1.cbl:exprocb1.pco
        $(ORACLE_HOME)\bin\procob release_cursor=yes $?
exprocb2.cbl:exprocb2.pco
        $(ORACLE_HOME)\bin\procob release_cursor=yes $?
# Link
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj \
             exprocb1.obj exprocb2.obj
        $(LK) -OutputFile $*.exe $** \
        [$(DCDIR)\spool\trnrmcmd\userobj\rm_obj.obj \]#3
        $(TP1LIB) $(SYSLIB) $(ORACLELIB)
```

#1

Specify `libdam.lib` only if you use TP1/FS/Direct Access.

#2

Specify `libtam.lib` only if you use TP1/FS/Table Access.

#3

If you use TP1/FS/Direct Access or TP1/FS/Table Access, add `OpenTP1_DAM` or `OpenTP1_TAM` when you use the `trnmkobj` command to create a transaction control object.

(8) Notes about linking Oracle

- If you link OpenTP1 and Oracle while using the X/Open XA interface, make sure

that all Oracle-related services are running.

- If you link OpenTP1 and Oracle while using the X/Open XA interface, use the following OpenTP1 functions to start transactions and acquire synchronization points:

  `dc_trn_begin`, `tx_begin`, `dc_trn_unchained_commit`, `dc_trn_chained_commit`, `tx_commit`, `dc_trn_unchained_rollback`, `dc_trn_chained_rollback`, `tx_rollback`

  If Oracle's SQL language is used to perform commit and rollback processing or if Oracle linked with the XA interface makes an attempt to use an unsupported function, transactions might not be processed correctly, resulting in a data integrity error in resources or in shutdown of the OpenTP1 system.

- Grant access permissions that enable a user with an OpenTP1 service logon account to write data to the Oracle troubleshooting information storage folder. If there is no write permission for this folder, OpenTP1 might not be able to store troubleshooting information for XA functions that are issued by OpenTP1 server processes.

- If you link OpenTP1 and Oracle while using the X/Open XA interface and either of the following messages is issued, make sure that Oracle and communication processes are running:

  - The `KFCA00901-W` message while Oracle's communication process (part of the SQL*Net function) is not running during transaction processing or results in an error.
  - Oracle's `ORA-1012` message during SQL access processing.

- If you link OpenTP1 and Oracle while using the X/Open XA interface and the number of processes that use the XA interface to access Oracle exceeds the value specified in Oracle's `SESSIONS` definition, the `KFCA00901-W` message is also issued. In such a case, use the following formula and, if necessary, revise Oracle's `SESSIONS` definition:

Minimum value of SESSIONS definition ≥ *basic number of Oracle accesses* x $n^{\#1}$

Basic number of Oracle accesses
≥ (*total number of user service processes that access Oracle*
+ *number of concurrent recovery processes*$^{\#2}$ + $1^{\#3}$)
x (*number of Oracle databases*)

#1

n indicates the number of sessions required to complete the processing from `xa_open` to `xa_start` to `UOC(SQL)` to `xa_end` to `xa_close` (normally about 1 to 5). For details, check the Oracle settings.

#2

The number of concurrent recovery processes is the `trn_recovery_process_count` operand value in the transaction service definition.

#3

`1` indicates the number of OpenTP1 resource manager monitoring processes.

- If you link OpenTP1 and Oracle while using the X/Open XA interface, Oracle's `SesTm` open string indicates the Oracle global transaction monitoring interval. If the value of `SesTm` is too small, Oracle might determine transactions without waiting for OpenTP1's instruction once the specified interval has elapsed, resulting in shutdown of the user server or of the OpenTP1 system. Be sure to specify an adequate value for `SesTm`.
- If you link OpenTP1 and Oracle while using the X/Open XA interface and Oracle is shut down and then restarted during transaction processing, Oracle might no longer accept requests from the process that had accessed Oracle before Oracle shut down. In such a case, the `KFCA00901-W`, `KFCA00905-E`, or `KFCA00906-E` message may be issued. If any of these messages is issued, terminate the corresponding server and then restart it. If the corresponding server is `_trnrcv`, terminate the OpenTP1 system and then restart it.

4.6.3 Linking with SQL Server

This subsection explains the settings required to link with SQL Server.

(1) Registering SQL Server

You must register SQL Server in the OpenTP1 system. Use the `trnlnkrm` command to do this.

The following shows an example of the `trnlnkrm` command:

```
trnlnkrm -a MS_SQL_Server -s msqlsrvxa1 -o xaswitch.lib
```

For details about the `trnlnkrm` command, see Chapter *7. Operation Commands*. For details about the information to be specified in the `trnlnkrm` command, such as RM name, RM switch name, and RM-related object name, see the SQL Server documentation.

The following notes explain execution of the `trnlnkrm` command.

- `-o` option in the `trnlnkrm` command

  Set the path of a related object in the `LIB` environment variable beforehand. If you specify the absolute path including folders for the related object, use `\\` as the folder delimiter.

- Specification of the prcsvpath definition command

  When you register SQL Server in the OpenTP1 system, the transaction service process uses SQL Server runtime libraries. Therefore, you must specify the folder containing the SQL Server runtime libraries in the prcsvpath process service definition command. The following shows an example of the prcsvpath definition command:

```
prcsvpath c:\mssql\binn
```

(2) Specifying the transaction service definition

Use the trnstring transaction service definition command to specify the information required in order to access the resource manager. When you link OpenTP1 with SQL Server, specify the -n, -o, -O, -c, and -C options and, if necessary, the -d option in the trnstring definition command.

Specify in the -o, -O, -c, and -C options the xa_open and xa_close function strings used to access the resource manager. Specify the xa_open and xa_close function strings as follows:

- For the -o, -O, -c, and -C options, specify the same value.
- For RmRecoveryGuid, specify a value that is unique globally. Specify the value (without the parentheses) acquired by Registry Format of the Windows guidgen utility program.
- For Timeout, specify the XA transaction timeout value (in milliseconds). When the specified timeout value expires, SQL Server (Windows MSDTC) rolls back the XA transaction automatically.

For details about the xa_open and xa_close function strings to be specified in SQL Server, see the SQL Server documentation.

The following shows an example of a transaction service definition:

```
set trn_extend_function=00000001    #add when SQL Server is used.

trnstring -n MS_SQL_Server \
-o "Tm=OpenTP1,\
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000" \
-O "Tm=OpenTP1,\
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000" \
-c "Tm=OpenTP1,\
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000" \
-C "Tm=OpenTP1,\
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000"
```

Note

If a character string enclosed in double quotation marks (") continues onto the next line following a continuation line symbol (\), make sure that the

specification on the continuation line begins in column 1.

Note:

Many of the errors that occur when an SQL Server database is accessed from OpenTP1 while using the X/Open XA interface are due to invalid `xa_open` and `xa_close` function string definitions. If the `KFCA00901-W` message is issued, check the `xa_open` and `xa_close` function string definitions and revise them as necessary.

(3) Compiling and linking UAPs

When you compile and link a UAP that accesses SQL Server, you must use the `trnmkobj` command to link the transaction control object created under `%DCDIR%\spool\trnrmcmd\userobj`.

For details about how to compile and link UAPs, see *3.2 Compiling and linking UAPs*, the edition of the manual *OpenTP1 Programming Reference* for the applicable language, and the SQL Server documentation. For details about the `trnmkobj` command, see Chapter *7. Operation Commands*.

The following shows examples of creating a transaction control object for UAPs that access SQL Server in C language and in COBOL.

In C language

```
trnmkobj -o rm_obj -r MS_SQL_Server
```

In COBOL

```
trnmkobj -o rm_obj -C "/Zl" -r MS_SQL_Server
```

(4) Example of a makefile (in C language)

This subsection presents an example of a makefile that is used when an SPP is created in C language.

(a) Source file names

The source file names used in the example makefile are as follows:

- Source program names of UAP created in C language:
  - `exmain.c` (main function)
  - `exsv1.c` (service function 1)
  - `exsv2.c` (service function 2)
  - `exsql.c` (SQL)
- RPC interface definition file name: `ex.def`

(b) Example of a makefile

The following shows an example of a makefile:

```
NODEBUG = 1
!include <ntwin32.mak>
#TP1TEST Make File
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
SQLSRVLIB = xaswitch.lib odbc32.lib

all:TP1TEST.exe
# Compile
exmain.obj:exmain.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsv1.obj:exsv1.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsv2.obj:exsv2.c
        $(cc) $(cflags) $(cvarsdll) $*.c
ex_sstb.obj:ex_sstb.c
        $(cc) $(cflags) $(cvarsdll) $*.c
exsql.obj:exsql.c
        $(cc) $(cflags) $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
        $(DCDIR)\bin\stbmake $?
# Link
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj exsql.obj
        $(link) $(conflags) -out:$*.exe $** \
        [$(DCDIR)\spool\trnrmcmd\userobj\rm_obj.obj \]#3
        $(TP1LIB) $(SYSLIB) $(SQLSRVLIB)
```

#1

Specify `libdam.lib` only if you use TP1/FS/Direct Access.

#2

Specify `libtam.lib` only if you use TP1/FS/Table Access.

#3

If you use TP1/FS/Direct Access or TP1/FS/Table Access, add `OpenTP1_DAM` or `OpenTP1_TAM` when you use the `trnmkobj` command to create a transaction control object.

(5) Example of a makefile (for COBOL)

This subsection presents an example of a makefile that is used when an SPP is created in COBOL.

(a) Source file names

The source file names used in the example makefile are as follows:

- Source program name of UAP created in COBOL2002: `main.cbl`

- Service program name of UAP created in COBOL2002: `sev1.cbl`
- Service program name of UAP created in COBOL2002: `sev2.cbl`
- SQL program name 1 created in COBOL2002: `exsql1.cbl`
- SQL program name 2 created in COBOL2002: `exsql2.cbl`
- RPC interface definition file name: `ex.def`

(b) Example of a makefile

The following shows an example of a makefile:

```
!include <ntwin32.mak>
# TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]#1 [libtam.lib]#2 libbetran.lib
SQLSRVOBJ = xaswitch.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
# Compile
main.obj:main.cbl
        $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
        $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
        $(CCBL) $(CBLFG2) sev2.cbl
exsql1.obj:exsql1.cbl
        $(CCBL) $(CBLFG2) exsql1.cbl
exsql2.obj:exsql2.cbl
        $(CCBL) $(CBLFG2) exsql2.cbl
ex_sstb.obj:ex_sstb.c
        $(cc) $(cflags) /Zl $(cvarsdll) $*.c
# Create a stub source file
ex_sstb.c:ex.def
        $(DCDIR)\bin\stbmake $?
# Link
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj \
             exsql1.obj exsql2.obj
        $(LK) -OutputFile $*.exe $** \
        [$(DCDIR)\spool\trnrmcmd\userobj\rm_obj.obj \]#3
        $(TP1LIB) $(SYSLIB) $(SQLSRVOBJ)
```

#1

Specify `libdam.lib` only if you use TP1/FS/Direct Access.

#2

Specify `libtam.lib` only if you use TP1/FS/Table Access.

#3

If you use TP1/FS/Direct Access or TP1/FS/Table Access, add `OpenTP1_DAM` or `OpenTP1_TAM` when you use the `trnmkobj` command to create a transaction control object.

(6) Notes about linking SQL Server

- If you link OpenTP1 and SQL Server while using the X/Open XA interface, make sure that all SQL Server-related services are running.
- If you link OpenTP1 and SQL Server while using the X/Open XA interface, use the following OpenTP1 functions to start transactions and acquire synchronization points:

  `dc_trn_begin`, `tx_begin`, `dc_trn_unchained_commit`, `dc_trn_chained_commit`, `tx_commit`, `dc_trn_unchained_rollback`, `dc_trn_chained_rollback`, `tx_rollback`

  If SQL Server's SQL language is used to perform commit and rollback processing or if SQL Server linked with the XA interface makes an attempt to use an unsupported function, transactions might not be processed correctly, resulting in a data integrity error in resources or in shutdown of the OpenTP1 system.
- The following notes apply to UAP creation when OpenTP1 and SQL Server are linked while using the X/Open XA interface.
  - Using DB-Library and XA interface to link OpenTP1 and SQL Server (creating UAPs in C or C++ language)

    In order to associate OpenTP1 transactions with an SQL Server connection while using the XA interface, you must issue a function (`dbenlistxatrans`) for establishing an association at the beginning of each transaction branch. If this function is not issued, the determination of transactions managed by OpenTP1 cannot be achieved. For details, see the SQL Server documentation.
  - Using an ODBC connection and the XA interface to link OpenTP1 and SQL Server (creating UAPs in C or C++ language)

    In order to associate OpenTP1 transactions with an SQL Server connection while using the XA interface, you must issue a function (`SQLSetconnectOption`) for establishing an association at the beginning of each transaction branch. If this function is not issued, the determination of transactions managed by OpenTP1 cannot be achieved. For details, see the SQL Server documentation.
  - Using an ODBC connection and the XA interface to link OpenTP1 and SQL Server (creating UAPs in COBOL2002)

    In order to associate OpenTP1 transactions with an SQL Server connection while using the XA interface, you must issue a service routine

(CBLSQLSETOPT provided by COBOL2002) for establishing association at the beginning of each transaction branch. If this service routine is not issued, the determination of transactions managed by OpenTP1 cannot be achieved. You must also specify the following environment variable in the user service definition of the user server that is linked with SQL Server by the XA interface:

```
putenv CBLSQLCOMMOD AUTO
```

- If you link OpenTP1 and SQL Server while using the X/Open XA interface, use of chained RPCs requires special attention. You must make sure that none of the three functions that associate OpenTP1 transactions with the SQL Server connection via the XA interface (dbenlistxatrans, SQLSetconnectOption, and CBLSQLSETOPT) is issued in the second or subsequent RPC.
- If you link OpenTP1 and SQL Server while using the X/Open XA interface, specify the Timeout setting for the XA interface open and close strings. If the Timeout setting is not specified and a UAP terminates abnormally, a transaction in the status SPID=-2 might remain permanently in SQL Server (Windows MSDTC). In Windows Server 2003, you must apply Service Pack 2 in order to specify the Timeout setting.
- If you link OpenTP1 and SQL Server while using the X/Open XA interface, specify the following setting in the transaction service definition at all OpenTP1 nodes associated with the transaction branch:

```
set trn_extend_function=00000001
```

If the following conditions are satisfied, OpenTP1's transaction determination function returns DCTRNER_HAZARD(-904) or TX_HAZARD(-4), not DC_OK(0):

- SQL Server is the only available resource manager access (one-phase commit is used to determine transactions).
- SQL Server rolls back transactions automatically based on the specified Timeout setting.

4.7 OS settings

This section explains the OS settings required in order to use OpenTP1 for Windows.

(1) TCP/IP settings

In an environment that uses multiple network interface cards (NICs), a list of IP addresses is returned in TCP/IP bind order. You must change the bind order so that the NIC used by OpenTP1 has top priority.

You change the bind order by choosing **Bind LAN** from **Control Panel**.

(2) Windows Firewall settings

If you specify Windows Firewall settings, enter the following commands from the command prompt:

```
for %p in ("%DCDIR%\bin\*.exe") do (netsh firewall add allowedprogram program="%p"
name="OpenTP1" scope=ALL)
for %p in ("%DCDIR%\lib\servers\*.exe") do (netsh firewall add allowedprogram
program="%p" name="OpenTP1" scope=ALL)
for %p in ("%DCDIR%\aplib\*.exe") do (netsh firewall add allowedprogram program="%p"
name="OpenTP1 UAP" scope=ALL)
```

The following notes apply to the Windows Firewall settings:

- The commands shown above cannot register a user service that is placed in a folder other than `%DCDIR%\aplib`. You must manually register such a user server in the exception list.
- If you execute the `trnlnkrm`, `rapsetup`, or `dcsetupml` command, the program might not be registered. If you have executed these commands, re-execute the applicable command shown above. If you have used the `dcsetupml` command to set up multiOpenTP1, execute the applicable command shown above from each multiOpenTP1's console used for command execution.
- If you add a user server, you must register it manually in the exception list.
- Use one of the following methods to register programs in the exception list:
  - **Exceptions** page that opens from **Windows Firewall** in **Control Panel**
  - Windows `netsh` command

    The following shows an example of the `netsh` command:

```
netsh firewall add allowedprogram program=c:\OpenTP1\aplib\sample.exe name="OpenTP1
UAP" scope=ALL
```

For details about the `netsh` command, see the Windows command Help.

(3) Computer name settings

Do not use `Administrators` as the full name of a computer on which OpenTP1 is run.

4.8 MultiOpenTP1 settings

This section explains the multiOpenTP1 settings in OpenTP1 for Windows that are different from OpenTP1 for UNIX. For the multiOpenTP1 settings that are common to OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Note:

In OpenTP1 for Windows, the OpenTP1 software originally installed by the installer is not considered to be a multiOpenTP1 environment.

(1) MultiOpenTP1 environment settings

This subsection explains how to set up multiOpenTP1.

- Setting up

  Use the `dcsetupml` command to set up all multiOpenTP1 environments. To set up a multiOpenTP1, you specify an identifier in the command argument. This identifier is used for registry, service, and event viewer registrations. The identifier is expressed as 1 to 4 single-byte alphanumeric characters.

  For details about the `dcsetupml` command, see *dcsetupml (sets up a multiOpenTP1 or cancels setup of a multiOpenTP1)* in Chapter *7. Operation Commands*.

  You can set up a maximum of three multiOpenTP1s. If you set up more than three multiOpenTP1s, operation is not guaranteed.

- Registering the service

  The identifier for a multiOpenTP1 specified in the `dcsetupml` command during setup is added to the end of the service name. For example, if the identifier is `M001`, the service name is registered as `OpenTP1_M001`. In this case, `OpenTP1_M001 - OpenTP1 for Windows` is displayed as the title at the OpenTP1 console.

- Message output to the Event Viewer

  When messages are output to the Event Viewer in a multiOpenTP1 environment, the identifier specified in the `dcsetupml` command during setup is added at the end to obtain the source name (in the same manner as with the service name). For example, if the identifier is `M001`, the source name is `OpenTP1_M001`. This source name identifies the OpenTP1 that issued the message.

(2) How to set up the multiOpenTP1 environment again

To set up the multiOpenTP1 environment again:

1. Make file backups in all the multiOpenTP1 environments.

   You must back up the following files:

   - System service information definition file (`%DCDIR%\lib\sysconf\mcfu*`)
   - System service common information definition file (`%DCDIR%\lib\sysconf\mcf`)
   - Communication service executable program (`%DCDIR%\lib\servers\mcfu*`)
   - Files under the multiOpenTP1 environment setup folder that have been changed and added by the user

2. Use the `dcsetupml -d` command to delete all multiOpenTP1 environments.
3. Install the replacement programs.
4. Use the `dcsetupml` command to set up all multiOpenTP1 environments.
5. In all multiOpenTP1 environments that have been set up, copy the backup files that were created in step 1 to overwrite the existing files.
6. If different resource managers[#] are to be linked to the installed environments, execute the `trnlnkrm` command for each environment.

   #

   Such resource managers as TP1/Message Control, TP1/FS/Direct Access, TP1/FS/Table Access, HiRDB, and Oracle.

Chapter

5. System Definitions

The OpenTP1 system definitions that can be specified differ between OpenTP1 for Windows and OpenTP1 for UNIX. This chapter provides lists of system definitions for purposes of describing the differences between OpenTP1 for Windows and OpenTP1 for UNIX. This chapter also provides detailed explanations of selected system definitions.

For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the OpenTP1 series manuals.

5.1 Lists of system definitions
5.2 Format of system definition explanations
5.3 Details of system definitions

5.1 Lists of system definitions

The system definitions that can be specified differ between OpenTP1 for Windows and OpenTP1 for UNIX. Limitations, such as unsupported arguments and settings, apply to some definitions. If you are a new user of OpenTP1 for Windows, we recommend that you use the lists provided in this section to check for differences from OpenTP1 for UNIX.

Once you have checked these lists, see the manual *OpenTP1 Version 7 System Definition* for those definitions common to OpenTP1 for Windows and OpenTP1 for UNIX. For those definitions that differ between them, see the applicable subsections in *5.3 Details of system definitions*, as necessary. For details about the definitions specific to OpenTP1 for Windows, see the applicable subsections in *5.3 Details of system definitions*.

Note:

The provided lists do not include the following definitions:

- Message queuing (TP1/Message Queue) definition
- TP1/NET/TCP/IP definition

  For details about this definition, see the manual *OpenTP1 Version 7 Protocol TP1/NET/TCP/IP*.
- TP1/Extension for .NET Framework definition

The legend for all the lists that follow is shown below.

Legend:

S: Same (definition is supported identically in OpenTP1 for Windows and OpenTP1 for UNIX)

W: Windows-specific (operand is specific to OpenTP1 for Windows)

D: Differences (definition has differences in OpenTP1 for Windows from OpenTP1 for UNIX or some limitations apply to OpenTP1 for Windows)

N: Not supported for OpenTP1 for Windows

--: Not applicable

(1) System environment definition

The following table lists and describes the system environment definition.

Table 5-1: System environment definition

Format	Operand/path	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	mode_conf	D	Only AUTO is supported as the setting. For details, see *System environment definition* in *5.3 Details of system definitions*.
	static_shmpool_size	S	--
	dynamic_shmpool_size	S	--
	shmpool_attribute	D	Only free is supported as the setting. For details, see *System environment definition* in *5.3 Details of system definitions*.
	user_command	D	Some notes apply. For details, see *System environment definition* in *5.3 Details of system definitions*.
	server_count	S	--
	user_server_ha	S	--
	system_terminate_watch_time	S	--
	start_scheduling_timing	S	--
	system_init_watch_time	S	--
	user_command_online	S	--
	preend_warning_watch_time	S	--
	user_command_online_tp1mngr_id	N	Specification of user ID is not supported.
	redirect_file	W	This operand is specific to OpenTP1 for Windows. For details, see *System environment definition* in *5.3 Details of system definitions*.
	redirect_file_name	W	
	redirect_file_size	W	
	console_output	W	
putenv	DCCONFPATH	S	--

Format	Operand/path	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	DCADMDEBUG	D	The default value differs from that for OpenTP1 for UNIX. For details, see *System environment definition* in *5.3 Details of system definitions*.
	DCUAPCONFPATH	S	--
dcputenv	DCCONFPATH	D	Some notes apply. For details, see *System environment definition* in *5.3 Details of system definitions*.
	DCUAPCONFPATH	D	

(2) System service configuration definition

The following table lists and describes the system service configuration definition.

Table 5-2: System service configuration definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	uap_conf		S	--
	dam_conf		S	--
	tam_conf		S	--
	que_conf		S	--
	ha_conf		S	--
	jar_conf		N	The global archive journal facility is not supported.
	mrs_conf		N	The remote MCF service is not supported.
	clt_conf		S	--
	ist_conf		N	The IST service is not supported.
	rmm_conf		N	The RMM service is not supported.
	xat_conf		N	The XATMI communication service is not supported.

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	mqa_conf		S	--
Command	dcsvstart	-m	S	--

(3) User service structure definition

The following table lists and describes the user service structure definition.

Table 5-3: User service structure definition

Format	Command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	dcsvstart	-u	S	--

(4) System common definition

The following table lists and describes the system common definition.

Table 5-4: System common definition

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rpc_trace		S	--
	rpc_trace_name		S	--
	rpc_trace_size		S	--
	name_port		S	--
	system_id		S	--
	all_node		D	Some notes apply. For details, see *System common definition* in *5.3 Details of system definitions*.
	node_id		S	--
	rpc_retry		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rpc_retry_count		S	--
	rpc_retry_interval		S	--
	multi_node_option		N	Multi-node (TP1/Multi) is not supported.
	prc_port		N	There is no need to specify this operand because multi-node (TP1/Multi) is not supported.
	rpc_delay_statistics		S	--
	my_host		S	--
	jp1_use		N	Event registration into JP1/Base's JP1 event service facility is not supported.
	rpc_message_level		S	--
	rpc_multi_tp1_in_same_host		N	An operation mode in which multiple OpenTP1s on the same host are used as the same global domain is not supported.
	max_socket_descriptors		S	--
	rpc_datacomp		S	--
	domain_masters_addr		N	RPC with a domain specified is not supported.
	domain_masters_port		N	
	domain_use_dns		N	
	client_uid_check		S	--
	rpc_port_base		S	--
	prf_trace		S	--
	trn_prf_trace_level		S	--
	core_suppress_watch_time		S	--
	rpc_netmask		S	--
	ipc_sockctl_highwater		S	--
	ipc_sockctl_watchtime		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	ipc_conn_interval		S	--
	ipc_send_interval		S	--
	ipc_send_count		S	--
	ipc_header_recv_time		S	--
	name_notify		D	Some notes apply. For details, see *System common definition* in *5.3 Details of system definitions*.
	all_node_ex		S	--
	rpc_server_busy_count		S	--
	rpc_send_retry_count		S	--
	rpc_send_retry_interval		S	--
	thdlock_sleep_time		S	--
	ipc_recvbuf_size		S	--
	ipc_sendbuf_size		S	--
	ipc_listen_sockbufset		S	--
	rpc_router_retry_count		S	--
	rpc_router_retry_interval		S	--
	ipc_backlog_count		S	--
	statistics		S	--
	name_domain_file_use		S	--
	all_node_extend_number		S	--
	all_node_ex_extend_number		S	--
	prc_current_work_path		S	--
	rpc_max_message_size		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	uap_trace_file_put		N	UAP trace files are output by default. For details about the UAP trace file output destination and file names, see the descriptions of the files output by OpenTP1 in the manual *OpenTP1 Version 7 Operation*.
	dcstart_wakeup_retry_count		S	--
	dcstart_wakeup_retry_interval		S	--
	nam_prf_trace_level		S	--
	fil_prf_trace_option		S	--
	fil_prf_trace_delay_time		S	--
	jnl_prf_event_trace_level		S	--
	jnl_fileless_option		N	The journal fileless mode is not supported.
	watch_time		S	--
Command	dcbindht	-h	S	--
		-n	S	--
putenv	LANG		S	--

(5) Lock service definition

The following table lists and describes the lock service definition.

Table 5-5: Lock service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	lck_limit_foruser	S	--
	lck_limit_fordam	S	--
	lck_limit_fortam	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	lck_limit_formqa	S	--
	lck_wait_timeout	S	--
	lck_deadlock_info	S	--
	lck_deadlock_info_remove	S	--
	lck_deadlock_info_remove_level	S	--
	lck_release_detect	S	--
	lck_release_detect_interval	S	--
	lck_prf_trace_level	S	--

(6) Timer service definition

The following table lists and describes the timer service definition.

Table 5-6: Timer service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	tim_watch_count	S	--

(7) Name service definition

The following table lists and describes the name service definition.

Table 5-7: Name service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	name_total_size	S	--
	name_cache_size	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	max_socket_descriptors	S	--
	name_global_lookup	S	--
	name_service_extend	S	--
	name_audit_conf	S	--
	name_audit_interval	S	--
	name_audit_watch_time	S	--
	name_rpc_control_list	S	--
	name_nodeid_check_message	S	--
	name_cache_validity_time	S	--
	watch_time	S	--

(8) Process service definition

The following table lists and describes the process service definition.

Table 5-8: Process service definition

Format	Operand/command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prc_process_count	S	--
	prc_recovery_resident	S	--
	prc_take_over_svpath	S	--
	term_watch_time	S	--
	max_socket_descriptors	S	--
	term_watch_count	S	--
	prc_prf_trace	S	--
	prc_coresave_path	S	--

Format	Operand/command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	prc_corecompress	S	--
	watch_time	S	--
Command	prcsvpath	D	Even if you change the path name for a user server, %DCDIR%\bin is always set at the beginning. For details, see *Process service definition (prcsvpath)* in *5.3 Details of system definitions*.

(9) Schedule service definition

The following table lists and describes the schedule service definition.

Table 5-9: Schedule service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	scd_server_count		S	--
	scd_hold_recovery		S	--
	scd_hold_recovery_count		S	--
	scd_port		S	--
	scd_this_node_first		S	--
	scd_announce_server_status		S	--
	max_socket_descriptors		S	--
	schedule_rate		S	--
	scd_retry_of_comm_error		S	--
	scd_advertise_control		S	--
	scd_message_level		S	--
	ipc_tcpnodelay		S	--
	watch_time		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	scdbufgrp	-g	N	Specification of schedule buffer groups is not supported.
		-n	N	
		-l	N	
	scdmulti	-m	S	--
		-p	S	--
		-g	S	--
		-t	S	--

(10) Transaction service definition

The following table lists and describes the transaction service definition.

Table 5-10: Transaction service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	trn_tran_process_count		S	--
	trn_recovery_process_count		S	--
	trn_expiration_time		S	--
	trn_expiration_time_suspend		S	--
	trn_tran_statistics		S	--
	trn_tran_recovery_list		S	--
	trn_cpu_time		N	Specification of a CPU monitoring interval is not supported. The value 0 (CPU time is not monitored) is assumed for this operand.
	trn_statistics_item		D	cputime is not supported as the setting.
	trn_max_subordinate_count		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	trn_rm_open_close_scope		S	--
	trn_optimum_item		S	--
	trn_processing_in_rm_error		S	--
	trn_recovery_list_remove		S	--
	trn_recovery_list_remove_level		S	--
	trn_crm_use		N	A communication resource manager is not supported.
	trn_max_crm_subordinate_count		N	
	trn_watch_time		S	--
	trn_rollback_information_put		S	--
	trn_limit_time		S	--
	trn_rollback_response_receive		S	--
	trn_partial_recovery_type		S	--
	max_socket_descriptors		S	--
	trn_recovery_failmsg_interval		S	--
	trn_wait_rm_open		S	--
	trn_retry_interval_rm_open		S	--
	trn_retry_count_rm_open		S	--
	thread_stack_size		S	--
	polling_control_data		S	--
	thread_yield_interval		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	groups		N	A group access list setting for a service group is not supported.
	trn_xar_use		S	--
	trn_start_recovery_mode		S	--
	trn_start_recovery_watch_time		S	--
	trn_start_recovery_interval		S	--
	trn_xa_commit_error		S	--
	trn_prf_event_trace_level		S	--
	trn_prf_event_trace_condition		S	--
	trn_completion_limit_time		S	--
	trn_extend_function		S	--
	watch_time		S	--
Command	trnstring	-n	S	--
		-i	S	--
		-o	S	--
		-c	S	--
		-O	S	--
		-C	S	--
		-d	S	--
		-e	S	--
		-m	S	--
		-r	S	--
putenv	Optional		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
dcputenv	Optional		S	--

(11) XA resource service definition

The following table lists and describes the XA resource service definition.

Table 5-11: XA resource service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	xar_eventtrace_level		S	--
	xar_eventtrace_record		S	--
	xar_session_time		S	--
	xar_msdtc_use		S	--
	xar_prf_trace_level		S	--
Command	xarfile	-t	S	--
		-a	S	--

(12) Interval service definition

The following table lists and describes the interval service definition.

Table 5-12: Interval service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	watch_time	S	--

(13) Status service definition

The following table lists and describes the status service definition.

Table 5-13: Status service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	sts_file_name_1	S	--
	sts_file_name_2	S	--
	sts_file_name_3	S	--
	sts_file_name_4	S	--
	sts_file_name_5	S	--
	sts_file_name_6	S	--
	sts_file_name_7	S	--
	sts_initial_error_switch	S	--
	sts_single_operation_switch	S	--
	sts_last_active_file	S	--
	sts_last_active_side	S	--
	watch_time	S	--

(14) Journal service definition

The following table lists and describes the journal service definition.

Table 5-14: Journal service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	jnl_tran_optimum_level		S	--
	jnl_arc_terminate_timeout		N	The global archive journal facility is not supported.
	max_socket_descriptors		S	--
	jnl_arc_ipc_buff_size		N	The global archive journal facility is not supported.

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	jnl_watch_time		D	The valid range includes the length of time to wait until a response to completion of journal file I/O processing is received. For details, see *Journal service definition* in *5.3 Details of system definitions*.
	watch_time		S	--
Command	jnldfsv	-r	S	--
		-c	S	--

(15) System journal service definition

The following table lists and describes the system journal service definition.

Table 5-15: System journal service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	jnl_max_datasize		S	--
	jnl_cdinterval		S	--
	jnl_rerun_swap		S	--
	jnl_dual		S	--
	jnl_singleoperation		S	--
	jnl_rerun_reserved_file_open		S	--
	jnl_arc_name		N	The global archive journal facility is not supported.
	jnl_arc_buff_size		N	
	jnl_arc_max_datasize		N	
	jnl_arc_terminate_check		N	
	jnl_arc_rec_kind		N	
	jnl_arc_uj_code		N	

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	jnl_arc_check_level		N	
	jnl_arc_trn_stat		N	
	jnl_unload_check		S	--
	jnl_auto_unload		S	--
	jnl_auto_unload_path		S	--
	jnl_max_file_dispersion		S	--
	jnl_min_file_dispersion		S	--
	watch_time		S	--
Command	jnladdfg	-g	S	--
		ONL	S	--
	jnladdpf	-g	S	--
		-e	S	--
		-a	S	--
		-b	S	--

(16) Checkpoint dump service definition

The following table lists and describes the checkpoint dump service definition.

Table 5-16: Checkpoint dump service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	jnl_objservername		S	--
	jnl_max_datasize		S	--
	assurance_count		S	--
	jnl_reduced_mode		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	jnl_reserved_file_auto_open		S	--
	jnl_dual		S	--
	jnl_singleoperation		S	--
	watch_time		S	--
Command	jnladdfg	-j	S	--
		-g	S	--
		ONL	S	--
	jnladdpf	-j	S	--
		-g	S	--
		-a	S	--
		-b	S	--

(17) Log service definition

The following table lists and describes the log service definition.

Table 5-17: Log service definition

Format	Operand/path	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	log_filesize	S	--
	log_msg_console	S	--
	log_msg_allno	S	--
	log_msg_prcid	S	--
	log_msg_prcno	S	--
	log_msg_sysid	S	--
	log_msg_date	S	--

Format	Operand/path	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	log_msg_time	S	--
	log_msg_hostname	S	--
	log_msg_pgmid	S	--
	log_netm_out	N	Network management by JP1/Cm2 is not supported.
	log_netm_allno	N	
	log_netm_prcid	N	
	log_netm_prcno	N	
	log_netm_sysid	N	
	log_netm_date	N	
	log_netm_time	N	
	log_netm_hostname	N	
	log_netm_pgmid	N	
	log_jp1_allno	N	Event registration into JP1/Base's JP1 event service facility is not supported.
	log_jp1_prcid	N	
	log_jp1_prcno	N	
	log_jp1_sysid	N	
	log_jp1_date	N	
	log_jp1_time	N	
	log_jp1_hostname	N	
	log_jp1_pgmid	N	
	log_notify_out	S	--
	log_notify_allno	S	--
	log_notify_prcid	S	--
	log_notify_prcno	S	--
	log_notify_sysid	S	--

Format	Operand/path	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	log_notify_date	S	--
	log_notify_time	S	--
	log_notify_hostname	S	--
	log_notify_pgmid	S	--
	log_jerr_rint	N	Event registration into JP1/Base's JP1 event service facility is not supported.
	log_syslog_out	N	Function for output of log information to syslog is not supported.
	log_syslog_allno	N	
	log_syslog_prcid	N	
	log_syslog_prcno	N	
	log_syslog_sysid	N	
	log_syslog_date	N	
	log_syslog_time	N	
	log_syslog_hostname	N	
	log_syslog_pgmid	N	
	log_syslog_append_nodeid	N	
	log_syslog_elist	N	
	log_syslog_elist_rint	N	
	log_syslog_synchro	N	
	log_audit_out	S	--
	log_audit_path	S	--
	log_audit_size	S	--
	log_audit_count	S	--
	log_audit_message	S	--
	watch_time	S	--

Format	Operand/path	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
putenv	TZ	S#	--
	DCSYSLOGOUT	N	Function for output of log information to syslog is not supported.

#

Before you specify the definition, check the notes on the TZ environment variable in *4.4 Environment variable settings*.

(18) Multi-node structure definition

The following table lists and describes the multi-node structure definition.

Table 5-18: Multi-node structure definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	dcmstart_watch_time		N	Multi-node (TP1/Multi) is not supported.
	dcmstop_watch_time		N	
	watch_time		N	
Command	dcmarea	-m	N	
		-g	N	
		-w	N	

(19) Multi-node physical definition

The following table lists and describes the multi-node physical definition.

Table 5-19: Multi-node physical definition

Format	Command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	dcprcport	-w	N	Multi-node (TP1/Multi) is not supported.
		-h	N	
		-p	N	

(20) Global archive journal service definition

The following table lists and describes the global archive journal service definition.

Table 5-20: Global archive journal service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	jnl_arc_terminate_timeout		N	The global archive journal facility is not supported.
	max_socket_descriptors		N	
	jnl_arc_ipc_buff_size		N	
	nl_watch_time		N	
Command	jnldfsv	-a	N	

(21) Archive journal service definition

The following table lists and describes the archive journal service definition.

Table 5-21: Archive journal service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	jnl_dual		N	The global archive journal facility is not supported.
	jnl_singleoperation		N	
	jnl_rerun_swap		N	

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	jnl_max_file_dispersion		N	
	jnl_min_file_dispersion		N	
	jnl_unload_check		N	
	jnl_arc_max_datasize		N	
Command	jnladdfg	-g	N	
		ONL	N	
	jnladdpf	-g	N	
		-e	N	
		-a	N	
		-b	N	

(22) DAM service definition

The following table lists and describes the DAM service definition.

Table 5-22: DAM service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	dam_update_block		S	--
	dam_added_file		S	--
	dam_update_block_over		S	--
	dam_message_level		S	--
	dam_tran_process_count		S	--
	dam_cache_size		S	--
	dam_cache_size_fix		S	--
	dam_cache_attribute		D	Only free is supported as the setting.

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	dam_io_interval		S	--
	dam_transaction_access		S	--
	dam_io_error_occur		S	--
	dam_cache_reuse_from		S	--
	dam_default_cache_num		S	--
	dam_ex_refer_read		S	--
	dam_max_block_size		S	--
	dam_kb_size		S	--
	watch_time		S	--
Command	damcache	Argument	S	--
	damchlmt	Argument	S	--
	damfile	-d	S	--
		-n	S	--
		-f	S	--
		-c	S	--
		Argument	S	--

(23) TAM service definition

The following table lists and describes the TAM service definition.

Table 5-23: TAM service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	tam_max_tblnum		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	tam_max_filesize		S	--
	tam_max_recsize		S	--
	tam_jnl_err_flag		S	--
	tam_pool_attri		D	Only free is supported as the setting.
	tam_tbl_lock_mode		S	--
	tam_cbl_level		S	--
	tam_max_trnnum		S	--
	tam_max_trnfilnum		S	--
	watch_time		S	--
Command	tamtable	-o	S	--
		-a	S	--
		-i	S	--
		-j	S	--
		Argument	S	--

(24) Client service definition

The following table lists and describes the client service definition.

Table 5-24: Client service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	parallel_count	S	--
	balance_count	S	--
	trn_expiration_time	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	trn_expiration_time_suspend	S	--
	trn_cpu_time	N	Specification of a CPU monitoring interval is not supported. The value 0 (CPU time is not monitored) is assumed for this operand.
	open_rm	S	--
	clt_inquire_time	S	--
	clt_port	S	--
	clt_trn_conf	S	--
	clt_cup_conf	S	--
	cup_parallel_count	S	--
	cup_balance_count	S	--
	clttrn_port	S	--
	cltcon_port	S	--
	trn_statistics_item	D	cputime is not supported as the setting.
	trn_optimum_item	S	--
	trn_watch_time	S	--
	trn_rollback_information_put	S	--
	trn_limit_time	S	--
	trn_rollback_response_receive	S	--
	trn_partial_recovery_type	S	--
	trn_completion_limit_time	S	--
	message_store_buflen	D	Some notes apply. For details, see *Client service definition* in *5.3 Details of system definitions*.
	watch_time	S	--

(25) IST service definition

The following table lists and describes the IST service definition.

Table 5-25: IST service definition

Format	Operand/command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	ist_node	N	The IST service is not supported.
	ist_node_group	N	
Command	istdef	N	

(26) RMM service definition

The following table lists and describes the RMM service definition.

Table 5-26: RMM service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rmm_check_services	N	The RMM service is not supported.
	rmm_system_behavior	N	
	rmm_down_with_system	N	
	rmm_sysdown_with_rm	N	

(27) Monitored RM definition

The following table lists and describes the monitored RM definition.

Table 5-27: Monitored RM definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rmm_start_command	N	The monitored RM definition is not supported.
	rmm_stop_command	N	

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rmm_abort_command	N	
	rmm_get_pid_command	N	
	rmm_command_watch_time	N	
	rmm_command_uid	N	
	rmm_command_gid	N	
	rmm_start_watch_time	N	

(28) Extended RM registration definition

The following table lists and describes the extended RM registration definition.

Table 5-28: Extended RM registration definition

Format	Command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	trnlnkrm	-a	N	Extended RM registration definition is not supported.
		-s	N	
		-o	N	
		-l	N	
		-f	N	

(29) XATMI communication service definition

The following table lists and describes the XATMI communication service definition.

Table 5-29: XATMI communication service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	xatinitapt		N	The XATMI communication service is not supported.
	xatinitaeq		N	
	xat_aso_con_event_svcname		N	
	xat_aso_discon_event_svcname		N	
	xat_aso_failure_event_svcname		N	
	max_open_fds		N	
	max_socket_descriptors		N	
Command	xatsrvadd	-p	N	
		-q	N	
		-s	N	

(30) MCF-message queue service definition

The following table lists and describes the MCF-message queue service definition.

Table 5-30: MCF-message queue service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	que_xidnum		S	--
	que_io_maxrecsize		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	quegrp	-g	S	--
		-f	S	--
		-n	S	--
		-m	S	--
		-w	S	--
		-c	S	--

(31) User service network definition

The following table lists and describes the user service network definition.

Table 5-31: User service network definition

Format	Command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	dcsvgdef	-g	S	--
		-h	S	--
		-p	S	--
		-t	S	--
		-w	S	--

(32) RAP-processing listener service definition

The following table lists and describes the RAP-processing listener service definition.

Table 5-32: RAP-processing listener service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rap_listen_port	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rap_parallel_server	D	Some notes apply. For details, see *RAP-processing listener service definition* in *5.3 Details of system definitions*.
	rap_watch_time	S	--
	rap_inquire_time	S	--
	nice	D	The meaning of this value differs from OpenTP1 for UNIX. For details, see *RAP-processing listener service definition* in *5.3 Details of system definitions*.
	uap_trace_max	S	--
	uid	N	Specification of user ID is not supported.
	rpc_response_statistics	S	--
	rpc_trace	S	--
	rpc_trace_name	S	--
	rpc_trace_size	S	--
	trn_expiration_time	S	--
	trn_expiration_time_suspend	S	--
	trn_cpu_time	N	Specification of a CPU monitoring interval is not supported. The value 0 (CPU time is not monitored) is assumed for this operand.
	trf_put	S	--
	trn_statistics_item	D	cputime is not supported as the setting.
	trn_optimum_item	S	--
	trn_watch_time	S	--
	trn_rollback_information_put	S	--
	trn_limit_time	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	trn_rollback_response_receive	S	--
	trn_partial_recovery_type	S	--
	rap_inquire_timeout_message	S	--
	rap_connection_assign_type	S	--
	rap_max_client	S	--
	rap_notify	S	--
	rap_client_manager_node	S	--
	rap_max_buff_size	S	--
	rap_io_retry_interval	S	--
	rap_sock_count	S	--
	rap_sock_interval	S	--
	rap_connect_retry_count	S	--
	rap_connect_retry_interval	S	--
	rap_listen_backlog	D	The value differs from that in OpenTP1 for UNIX. For details, see *RAP-processing listener service definition* in *5.3 Details of system definitions*.
	rap_msg_output_interval	S	--
	rap_recovery_server	S	--
	rap_connect_interval	S	--
	rpc_extend_function	S	--
	max_socket_descriptors	S	--
	trn_completion_limit_time	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rap_message_id_change_level	S	--
	rap_term_disconnect_time	S	--
	rap_stay_watch_time	S	--
	rap_stay_warning_interval	S	--
	log_audit_out_suppress	S	--
	log_audit_message	S	--
	ipc_sockctl_highwater	S	--
	ipc_sockctl_watchtime	S	--
	watch_time	S	--

(33) RAP-processing client manager service definition

The following table lists and describes the RAP-processing client manager service definition.

Table 5-33: RAP-processing client manager service definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rap_client_manager_port	S	--
	rap_listen_inf	S	--
	uid	N	Specification of user ID is not supported.
	log_audit_out_suppress	S	--
	log_audit_message	S	--
	rap_watch_time	S	--

(34) Definition of performance verification trace

The following table lists and describes the definition of performance verification trace.

Table 5-34: Definition of performance verification trace

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prf_file_size	S	--
	prf_information_level	D	Messages related to performance verification traces are output to the Event Viewer. This output of messages to the Event Viewer cannot be suppressed because the DCSYSLOGOUT environment variable is not supported in the system common definition or the definition of performance verification trace.
	prf_file_count	S	--
	prf_trace_backup	S	--

(35) XAR performance verification trace definition

The following table lists and describes the XAR performance verification trace definition.

Table 5-35: XAR performance verification trace definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prf_file_size	S	--
	prf_information_level	D	Messages related to XAR performance verification traces are output to the Event Viewer. This output of messages to the Event Viewer cannot be suppressed because the DCSYSLOGOUT environment variable is not supported in the system common definition or the definition of performance verification trace.
	prf_file_count	S	--

(36) JNL performance verification trace definition

The following table lists and describes the JNL performance verification trace definition.

Table 5-36: JNL performance verification trace definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prf_file_size	S	--
	prf_file_count	S	--
	prf_trace_backup	S	--

(37) LCK performance verification trace definition

The following table lists and describes the LCK performance verification trace definition.

Table 5-37: LCK performance verification trace definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prf_file_size	S	--
	prf_information_level	D	Messages related to LCK performance verification traces are output to the Event Viewer. This output of messages to the Event Viewer cannot be suppressed because the DCSYSLOGOUT environment variable is not supported in the system common definition or the definition of performance verification trace.
	prf_file_count	S	--

(38) TRN event trace definition

The following table lists and describes the TRN event trace definition.

Table 5-38: TRN event trace definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prf_file_size	S	--

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	prf_information_level	D	Messages related to TRN event traces are output to the Event Viewer. This output of messages to the Event Viewer cannot be suppressed because the DCSYSLOGOUT environment variable is not supported in the system common definition or the definition of performance verification trace.
	prf_file_count	S	--

(39) Real-time statistics service definition

The following table lists and describes the real-time statistics service definition.

Table 5-39: Real-time statistics service definition

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rts_trcput_interval		S	--
	rts_service_max		S	--
	rts_item_max		S	--
	rts_log_file		S	--
	rts_log_file_name		S	--
	rts_log_file_size		S	--
	rts_log_file_count		S	--
	rts_log_file_backup		S	--
	rts_swap_message		S	--

Format	Operand/ command	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	rtsput	-u	S	--
		-s	S	--
		-v	S	--
		-o	S	--
		-b	S	--
		-e	S	--
		-f	S	--

(40) Real-time acquisition item definition

The following table lists and describes the real-time acquisition item definition.

Table 5-40: Real-time acquisition item definition

Format	Operand	Definition type	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	rts_cpd_collct_cpd	Checkpoint dump information	S	--
	rts_cpd_validt_cpd		S	--
	rts_jnl_buf_full	Journal information	S	--
	rts_jnl_wait_buf		S	--
	rts_jnl_jnl_output		S	--
	rts_jnl_io_wait		S	--
	rts_jnl_write		S	--
	rts_jnl_swap		S	--
	rts_jnl_jnl_input		S	--
	rts_jnl_read		S	--
	rts_lck_lock_acqst	Lock information	S	--
	rts_lck_lock_wait		S	--

Format	Operand	Definition type	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rts_lck_deadlock		S	--
	rts_nam_global_cache_hit	Name information	S	--
	rts_nam_local_cache_hit		S	--
	rts_nam_lookup		S	--
	rts_nam_node_lookup		S	--
	rts_nam_node_lookup_responce		S	--
	rts_osl_stamem_acq	Shared memory management information	S	--
	rts_osl_stamem_pol		S	--
	rts_osl_dynmem_acq		S	--
	rts_osl_dynmem_pol		S	--
	rts_prc_prc_genert	Process information	S	--
	rts_prc_uap_abnml		S	--
	rts_prc_sys_abnml		S	--
	rts_prc_prc_term		S	--
	rts_prc_prc_num		S	--
	rts_que_read	Message queue information	S	--
	rts_que_write		S	--
	rts_que_read_err		S	--
	rts_que_write_err		S	--
	rts_que_wait_buf		S	--
	rts_que_real_read		S	--
	rts_que_real_write		S	--
	rts_que_delay_wrt		S	--
	rts_que_delay_rec		S	--
	rts_que_delay_msg		S	--

Format	Operand	Definition type	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rts_rpc_rpc_call	RPC information	S	--
	rts_rpc_rpc_call_chained		S	--
	rts_rpc_usr_srvc		S	--
	rts_rpc_rpc_ovrtim		S	--
	rts_scd_scd_wait	Schedule information	S	--
	rts_scd_schedule		S	--
	rts_scd_using_buf		S	--
	>rts_scd_lack_buf		S	--
	rts_scd_scd_stay		S	--
	rts_scd_svc_scd_wait		S	--
	rts_scd_svc_using_buf		S	--
	rts_scd_parallel		S	--
	rts_trn_commit	Transaction information	S	--
	rts_trn_rollback		S	--
	rts_trn_cmt_cmd		S	--
	rts_trn_rbk_cmd		S	--
	rts_trn_haz_cmd		S	--
	rts_trn_mix_cmd		S	--
	rts_trn_branch		S	--
	rts_trn_sync_point		S	--
	rts_dam_read	DAM information	S	--
	rts_dam_read_err		S	--
	rts_dam_write		S	--
	rts_dam_write_err		S	--
	rts_dam_fj		S	--
	rts_dam_trn_branch		S	--

Format	Operand	Definition type	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rts_dam_cache_block		S	--
	rts_dam_shm_pool		S	--
	rts_tam_real_renew	TAM information	S	--
	rts_tam_real_renew_time		S	--
	rts_tam_rec_refer		S	--
	rts_tam_rec_renew		S	--
	rts_tam_read		S	--
	rts_tam_read_err		S	--
	rts_tam_write		S	--
	rts_tam_write_err		S	--
	rts_xar_start	XA resource service information	S	--
	rts_xar_start_err		S	--
	rts_xar_call		S	--
	rts_xar_call_err		S	--
	rts_xar_end		S	--
	rts_xar_end_err		S	--
	rts_xar_prepare		S	--
	rts_xar_prepare_err		S	--
	rts_xar_commit		S	--
	rts_xar_commit_err		S	--
	rts_xar_rollback		S	--
	rts_xar_rollback_err		S	--
	rts_xar_recover		S	--
	rts_xar_recover_err		S	--
	rts_xar_forget		S	--
	rts_xar_forget_err		S	--

Format	Operand	Definition type	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rts_mcf_ap_scd_stay	MCF information	S	--
	rts_mcf_ap_usr_srvc		S	--
	rts_mcf_in_msg_scd_wait		S	--
	rts_mcf_out_msg_sync_scd_wait		S	--
	rts_mcf_out_msg_resp_scd_wait		S	--
	rts_mcf_out_msg_prio_scd_wait		S	--
	rts_mcf_out_msg_norm_scd_wait		S	--
	rts_mcf_que_scd_wait_num		S	--

(41) User service default definition

The following table lists and describes the user service default definition.

Table 5-41: User service default definition

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	nice		D	The meaning of this value differs from OpenTP1 for UNIX. For details, see *User service default definition* in *5.3 Details of system definitions*.
	parallel_count		S	--
	hold		S	--
	hold_recovery		S	--
	deadlock_priority		S	--
	schedule_priority		S	--
	message_buflen		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	message_store_buflen		D	Some notes apply. For details, see *User service default definition* in *5.3 Details of system definitions*.
	trn_expiration_time		S	--
	trn_expiration_time_suspend		S	--
	watch_next_chain_time		S	--
	atomic_update		S	--
	receive_from		S	--
	uap_trace_max		S	--
	uap_trace_file_put		N	UAP trace files are output by default. For details about the UAP trace file output destination and file names, see the descriptions of the files output by OpenTP1 in the manual *OpenTP1 Version 7 Operation*.
	term_watch_time		S	--
	mcf_jnl_buff_size		S	--
	type		S	--
	balance_count		S	--
	uid		N	Specification of user ID is not supported.
	auto_restart		S	--
	critical		S	--
	lck_wait_priority		S	--
	mcf_psv_id		S	--
	trn_cpu_time		N	Specification of a CPU monitoring interval is not supported. The value 0 (CPU time is not monitored) is assumed for this operand.
	service_hold		S	--
	service_priority_control		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	message_cell_size		D	Some notes apply. For details, see *User service default definition* in *5.3 Details of system definitions*.
	max_socket_msg		S	--
	max_socket_msglen		S	--
	trf_put		S	--
	mcf_mgrid		S	--
	mcf_service_max_count		S	--
	trn_statistics_item		D	cputime is not supported as the setting.
	node_down_restart		S	--
	rpc_response_statistics		S	--
	server_type		D	Only betran is supported as the setting.
	trn_rm_open_close_scope		S	--
	trn_optimum_item		S	--
	purge_msgget		S	--
	cancel_normal_terminate		S	--
	prc_abort_signal		D	Specify 3, 6, or 15 as the signal number. For details, see *User service default definition* in *5.3 Details of system definitions*.
	rpc_service_retry_count		S	--
	rpc_extend_function		S	--
	max_socket_descriptors		S	--
	max_open_fds		S	--
	service_term_watch_time		S	--
	termed_after_service		S	--
	xat_trn_expiration_time		N	The XATMI communication service is not supported.

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	xat_osi_usr		N	
	rpc_trace		S	--
	rpc_trace_name		S	--
	rpc_trace_size		S	--
	trn_rollback_information_put		S	--
	schedule_method		N	Specification of a user server scheduling method is not supported.
	service_wait_time		N	
	mcf_spp_oj		N	Specification of whether acquisition of SPP log information OJ is required is not supported.
	adm_message_option		S	--
	trn_watch_time		S	--
	trn_limit_time		S	--
	trn_rollback_response_receive		S	--
	trn_partial_recovery_type		S	--
	rpc_destination_mode		S	--
	rpc_rap_auto_connect		S	--
	rpc_rap_inquire_time		S	--
	rpc_request_cancel_for_timedout		S	--
	status_change_when_termin g		S	--
	service_expiration_time		S	--
	multi_schedule		S	--
	make_queue_on_starting		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	loadcheck_interval		S	--
	levelup_queue_count		S	--
	leveldown_queue_count		S	--
	ipc_sockctl_highwater		S	--
	ipc_sockctl_watchtime		S	--
	ipc_conn_interval		S	--
	ipc_send_interval		S	--
	ipc_send_count		S	--
	ipc_header_recv_time		S	--
	rpc_send_retry_count		S	--
	rpc_send_retry_interval		S	--
	ipc_recvbuf_size		S	--
	ipc_sendbuf_size		S	--
	ipc_listen_sockbufset		S	--
	polling_control_data		S	--
	thread_yield_interval		S	--
	groups		N	A group access list setting for service groups is not supported.
	loadlevel_message		S	--
	ipc_backlog_count		S	--
	rpc_buffer_pool_max		S	--
	schedule_delay_limit		S	--
	schedule_delay_abort		S	--
	rap_autoconnect_con_error_msg		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	core_shm_suppress		N	Shared memory dumps are not output to the core file.
	xat_connect_resp_time		N	The XATMI communication service is not supported.
	scd_poolfull_check_interval		S	--
	scd_poolfull_check_count		S	--
	scd_pool_warning_use_rate		S	--
	scd_pool_warning_interval		S	--
	ipc_tcpnodelay		S	--
	stay_watch_queue_count		S	--
	stay_watch_check_rate		S	--
	stay_watch_abort		S	--
	stay_watch_start_interval		S	--
	stay_watch_check_interval		S	--
	trn_completion_limit_time		S	--
	rap_message_id_change_level		S	--
	log_audit_out_suppress		S	--
	log_audit_message		S	--
	mcf_prf_trace		S	--
	watch_time		S	--
	process_privilege_restrict		W	This operand is specific to OpenTP1 for Windows. For details, see *User service default definition* in *5.3 Details of system definitions*.
	process_privilege_name		W	
Command	trnrmid	-n	S	--
		-i	S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	scdbufgrp	-g	N	Specification of schedule buffer groups is not supported.
	scdmulti	-g	S	--
	scdsvcdef	-c	S	--
		-p	S	--
		-n	S	--
		-l	S	--
putenv	Optional		D	Some notes apply. For details, see *User service default definition* in *5.3 Details of system definitions*.
	XAT_CONNECT_RESP_TIME		N	The XATMI communication service is not supported.
dcputenv	Optional		D	Some notes apply. For details, see *User service default definition* in *5.3 Details of system definitions*.

(42) User service definition

The following table lists and describes the user service definition.

Table 5-42: User service definition

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	service_group		S	--
	module		D	Some notes apply. For details, see *User service definition* in *5.3 Details of system definitions*.
	service		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	nice		D	The meaning of this value differs from OpenTP1 for UNIX. For details, see *User service definition* in *5.3 Details of system definitions*.
	parallel_count		S	--
	hold		S	--
	hold_recovery		S	--
	deadlock_priority		S	--
	schedule_priority		S	--
	message_buflen		S	--
	message_store_buflen		D	Some notes apply. For details, see *User service definition* in *5.3 Details of system definitions*.
	trn_expiration_time		S	--
	trn_expiration_time_suspend		S	--
	watch_next_chain_time		S	--
	atomic_update		S	--
	receive_from		S	--
	uap_trace_max		S	--
	uap_trace_file_put		N	UAP trace files are output by default. For details about the UAP trace file output destination and file names, see the descriptions of files output by OpenTP1 in the manual *OpenTP1 Version 7 Operation*.
	term_watch_time		S	--
	mcf_jnl_buff_size		S	--
	type		S	--
	balance_count		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	uid		N	Specification of user ID is not supported.
	auto_restart		S	--
	critical		S	--
	lck_wait_priority		S	--
	mcf_psv_id		S	--
	trn_cpu_time		N	Specification of a CPU monitoring interval is not supported. The value 0 (CPU time is not monitored) is assumed for this operand.
	service_hold		S	--
	service_priority_control		S	--
	message_cell_size		D	Some notes apply. For details, see *User service definition* in *5.3 Details of system definitions*.
	max_socket_msg		S	--
	max_socket_msglen		S	--
	trf_put		S	--
	mcf_mgrid		S	--
	mcf_service_max_count		S	--
	trn_statistics_item		D	cputime is not supported as the setting.
	node_down_restart		S	--
	rpc_response_statistics		S	--
	server_type		D	Only betran is supported as the setting.
	trn_rm_open_close_scope		S	--
	trn_optimum_item		S	--
	purge_msgget		S	--
	cancel_normal_terminate		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	prc_abort_signal		D	Specify 3, 6, or 15 as the signal number. For details, see *User service definition* in *5.3 Details of system definitions*.
	rpc_service_retry_count		S	--
	rpc_extend_function		S	--
	max_socket_descriptors		S	--
	max_open_fds		S	--
	service_term_watch_time		S	--
	termed_after_service		S	--
	xat_trn_expiration_time		N	The XATMI communication service is not supported.
	xat_osi_usr		N	
	rpc_trace		S	--
	rpc_trace_name		S	--
	rpc_trace_size		S	--
	trn_rollback_information_put		S	--
	schedule_method		N	Specification of a user server scheduling method is not supported.
	service_wait_time		N	
	mcf_spp_oj		N	Specification of whether acquisition of SPP log information OJ is required is not supported.
	adm_message_option		S	--
	trn_watch_time		S	--
	trn_limit_time		S	--
	trn_rollback_response_receive		S	--
	trn_partial_recovery_type		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rpc_destination_mode		S	--
	rpc_rap_auto_connect		S	--
	rpc_rap_inquire_time		S	--
	rpc_request_cancel_for_timedout		S	--
	status_change_when_termin g		S	--
	service_expiration_time		S	--
	multi_schedule		S	--
	make_queue_on_starting		S	--
	loadcheck_interval		S	--
	levelup_queue_count		S	--
	leveldown_queue_count		S	--
	ipc_sockctl_highwater		S	--
	ipc_sockctl_watchtime		S	--
	ipc_conn_interval		S	--
	ipc_send_interval		S	--
	ipc_send_count		S	--
	ipc_header_recv_time		S	--
	rpc_send_retry_count		S	--
	rpc_send_retry_interval		S	--
	ipc_recvbuf_size		S	--
	ipc_sendbuf_size		S	--
	ipc_listen_sockbufset		S	--
	polling_control_data		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	thread_yield_interval		S	--
	groups		N	A group access list setting for service groups is not supported.
	loadlevel_message		S	--
	ipc_backlog_count		S	--
	rpc_buffer_pool_max		S	--
	schedule_delay_limit		S	--
	schedule_delay_abort		S	--
	rap_autoconnect_con_error_msg		S	--
	core_shm_suppress		N	Shared memory dumps are not output to the core file.
	xat_connect_resp_time		N	The XATMI communication service is not supported.
	scd_poolfull_check_interval		S	--
	scd_poolfull_check_count		S	--
	scd_pool_warning_use_rate		S	--
	scd_pool_warning_interval		S	--
	ipc_tcpnodelay		S	--
	stay_watch_queue_count		S	--
	stay_watch_check_rate		S	--
	stay_watch_abort		S	--
	stay_watch_start_interval		S	--
	stay_watch_check_interval		S	--
	trn_completion_limit_time		S	--

Format	Operand/ command/path	Option	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	rap_message_id_change_level		S	--
	log_audit_out_suppress		S	--
	log_audit_message		S	--
	mcf_prf_trace		S	--
	watch_time		S	--
	process_privilege_restrict		W	This operand is specific to OpenTP1 for Windows. For details, see *User service definition* in *5.3 Details of system definitions*.
	process_privilege_name		W	
Command	trnrmid	-n	S	--
		-i	S	--
	scdbufgrp	-g	N	Specification of schedule buffer groups is not supported.
	scdmulti	-g	S	--
	scdsvcdef	-c	S	--
		-p	S	--
		-n	S	--
		-l	S	--
putenv	Optional		D	Some notes apply. For details, see *User service definition* in *5.3 Details of system definitions*.
	DCFPL_CONNECT_RETRY_COUNT		S	--
	DCFPL_CONNECT_RETRY_INTERVAL		S	--
	XAT_CONNECT_RESP_TIME		N	The XATMI communication service is not supported.
dcputenv	Optional		D	Some notes apply. For details, see *User service definition* in *5.3 Details of system definitions*.

(43) MCF manager definition

The following table lists and describes the MCF manager definition.

Table 5-43: MCF manager definition

Format	Command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	mcfmenv	S	--
	mcfmcomn	D	The cmdsvname operand of the -o option (MCF online command service) and the stats operand of the -w option (MCF statistics) are not supported.
	mcfmcname	S	--
	mcfmuap	S	--
	mcfmqgid	S	--
	mcfmexp	S	--
	mcfmsts	S	--
	mcfmsmsg	S	--
	mcfmsvg	S	--

(44) MCF communication configuration definition

The following table lists and describes the MCF communication configuration definition.

Table 5-44: MCF communication configuration definition

Format	Command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	mcftenv	D	The diskitq operand of the -q option (high-speed MCF start processing when memory queue is used for input queue) is not supported.
	mcftcomn	S	--
	mcfttred	S	--

Format	Command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	mcfttim	S	--
	mcfttrc	S	--
	mcftsts	S	--
	mcftbuf	S	--
	mcftpsvr	S	--
	mcftalcle	S	--
	mcftped	S	--

(45) MCF application definition

The following table lists and describes the MCF application definition.

Table 5-45: MCF application definition

Format	Command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Command	mcfaenv	S	--
	mcfaalcap	D	SPP (SPP start from MCF) is not supported as the type operand value in the -g option.

(46) MCF performance verification trace definition

The following table lists and describes the MCF performance verification trace definition.

Table 5-46: MCF performance verification trace definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	prf_file_size	S	--
	prf_file_count	S	--

(47) Definition of system service information

The following table lists and describes the definition of system service information.

Table 5-47: Definition of system service information

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	module	D	Some notes apply. For details, see *Definition of system service information* in *5.3 Details of system definitions*.
	mcf_prf_trace	S	--

(48) System service common information definition

The following table lists and describes the system service common information definition.

Table 5-48: System service common information definition

Format	Operand	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
set	max_socket_descriptors	S	--
	max_open_fds	S	--
	thdlock_sleep_time	S	--
	mcf_prf_trace_level	S	--

5.2 Format of system definition explanations

This section explains the format used to describe a definition (definition command).

definition-name (definition-command)

Formats

Explains the formats of options and operands that are specific to OpenTP1 for Windows and that differ from OpenTP1 for UNIX.

Description or options

Operands in the set format

Explains the definition operands.

Command format

Explains a definition that can be specified in command format (definition command).

Operands in the putenv format

Explains the `putenv` format that is used to specify a user's environment variables.

Operands in the dcputenv format

Explains the `dcputenv` format that is used to specify a user's environment variables.

Notes

Provides notes about the command.

5.3 Details of system definitions

This section provides details about the definitions in OpenTP1 for Windows that differ from those in OpenTP1 for UNIX. For details about the definitions (operands) that are not explained here, see the lists in *5.1 Lists of system definitions*.

System environment definition

For details about the operands that are not explained here, see the list in *5.1(1) System environment definition*.

Formats

- `set` format

```
[set mode_conf=AUTO]
[set shmpool_attribute=free]
[set user_command=user-environment-setting-command]
[set redirect_file=Y|N]
[set redirect_file_name=output-file-name]
[set redirect_file_size=maximum-size-of-output-file]
[set console_output=Y|N]
```

- `putenv` format

```
[putenv DCADMDEBUG 0|1]
```

- `dcputenv` format

```
[dcputenv DCCONFPATH definition-file-storage-folder]
[dcputenv DCUAPCONFPATH user-service-definition-file-storage-folder]
```

Description

Operands in the set format

- `mode_conf=AUTO`

  ~<<AUTO>>

  Only `AUTO` can be specified as the system startup method. For differences in the startup mode depending on the mode of the previous termination, see *6.2.4 Determining the startup mode*.

  There are no other differences.

- `shmpool_attribute=free`

  ~<<free>>

  `fixed` is not supported because a shared memory pool for system services cannot be fixed in memory. There are no other differences.

- `user_command=`*user-environment-setting-command*

  ~<path name>

  Specifies the complete path name of the user environment setting command that is to be executed before OpenTP1 is started.

  Do not specify a command or batch file that will cause a default input wait status.

  There are no other differences.

- `redirect_file=Y|N`

  ~<<N>>

  This operand is specific to OpenTP1 for Windows. It specifies whether the contents of standard output and standard error issued from processes under OpenTP1 are to be output to a file (whether the standard output redirect facility is to be used).

  `Y`

  Output the contents of standard output and standard error to a file. The OpenTP1 console window is not displayed in such a case.

  `N`

  Do not output the contents of standard output and standard error to a file.

  The standard output redirect facility cannot be used when you specify the `prc_port` or `rpc_port_base` operand in the system common definition. For details, see *2.2.4 Relationship among system definitions*.

  If you specify `Y` in this operand when you have specified the `prc_port` or `rpc_port_base` operand, OpenTP1 issues the `KFCA26520-W` message, ignores this operand, and resumes startup of OpenTP1. When the `KFCA26520-W` message is issued, either delete the `prc_port` or `rpc_port_base` operand or delete the `redirect_file` operand.

- `redirect_file_name=`*output-file-name*

  ~<path name>

  This operand is specific to OpenTP1 for Windows. It specifies the absolute path name of the file to which the contents of standard output and standard error are to be output.

  If file generations are managed, a generation number (`1` or `2`) is added at the end of the file name. If the file name is omitted, the two files `%DCDIR%\spool\prclog1` and `%DCDIR%\spool\prclog2` are created.

- `redirect_file_size=`*maximum-size-of-output-file*

  ~<unsigned integer>((0 to 65535))<<1024>>(kilobytes)

  This operand is specific to OpenTP1 for Windows. It specifies (in kilobytes) the

maximum size of the file to which the contents of standard output and standard error are to be output.

Two generations of output files are managed based on the specified file size. The size of an output file might exceed the specified value depending on the messages that are output.

If 0 is specified, the output file will be an incremental file that increases monotonically because the file is not managed as having generations.

If you use the standard output redirect facility and output a large number of messages concurrently, the two output files might be updated at the same time and the most recent file generation might be assigned as the current file the next time OpenTP1 starts. Therefore, as a guideline, specify the size of one generation of file that takes at least one second to fill up.

- console_output=Y|N

  ~<<N>>

This operand is specific to OpenTP1 for Windows. It specifies whether the contents of standard output and standard error are to be output to the console (whether the OpenTP1 console output facility is to be used).

Y

Output the contents of standard output and standard error to the console.

N

Do not output the contents of standard output and standard error to the console.

Even when Y is specified in this operand, the contents of standard output and standard error might not be output to the console, depending on the values set in other operands. For details, see *2.2.4 Relationship among system definitions*.

Operands in the putenv format

- DCADMDEBUG 0|1

  ~<<1>>

The default value is 1. There are no other differences.

Operands in the dcputenv format

- DCCONFPATH *definition-file-storage-folder*

  ~<path name>

The value of the DCCONFPATH environment variable is case sensitive. Make sure that the specified character strings are identical throughout the computer.

Enclose each environment variable in percent signs (%), not dollar signs ($). There are

no other differences.

- `DCUAPCONFPATH` *user-service-definition-file-storage-folder*

  ~<path name>

  Enclose each environment variable in percent signs (`%`), not dollar signs (`$`). There are no other differences.

System common definition

For details about the operands that are not explained here, see the list in *5.1(4) System common definition*.

Formats

- set format

```
[set all_node="node-name[:port-number][:high]"
              [,"node-name[:port-number][:high]"...]]
[set name_notify=Y|N]
```

Description

Operands in the set format

- `all_node="node-name[:port-number][:high]"[,"node-name[:port-number][:high]"...]`

  You must specify the node name that is specified in this operand in `%SystemRoot%\system32\drivers\etc\HOSTS` together with the IP address beforehand or use the WINS server. There are no other differences.

- `name_notify=Y|N`

  ~<<N>>

  If `Y` is specified in this operand, OpenTP1 performs start notification processing as many times as there are nodes specified in the `all_node` operand.

  Note that when you specify `Y` in this operand and the specified `all_node` operand value includes nodes at which OpenTP1 is not running, the following time is required before the inactive OpenTP1 systems are placed in online status:

  When the `all_node` operand value includes nodes that result in a timeout during `connect()` processing:

  Number of nodes resulting in timeout x *ipc_conn_interval operand value in the system common definition* x 2 (seconds)

  When the `all_node` operand value includes nodes to which OpenTP1 cannot send notification:

  Number of nodes to which notification cannot be sent x *TCP/IP re-forwarding processing time*[#] x 2 (seconds)

  #

  This is the time required for TCP/IP re-forwarding processing, as determined

by registry parameters (such as `TcpMaxConnectRetransmissions`). It is about one second in the default settings.

Process service definition (prcsvpath)

For details about the operands that are not explained here, see the list in *5.1(8) Process service definition*.

Formats

[prcsvpath *path-name*]

Options

- *path-name*

  ~<path name><<%DCDIR%\aplib;%DCDIR%\bin>>

  The differences from OpenTP1 for UNIX are as follows (there are no other differences):

  - Even if you change the path name of the user server, %DCDIR%\bin is always set at the beginning.
  - The path name cannot contain any spaces. If you want to specify a path name containing a space, first convert it to a short name in 8.3 format (MS-DOS format).
  - If you use XMAP3 Server Runtime Version 5, set *XMAP3-Server-Runtime-installation-folder*\bin.

  For details about the path names of runtime libraries, see *4.3.3 Setting the runtime library path*.

Journal service definition

For details about the operands that are not explained here, see the list in *5.1(14) Journal service definition*.

Formats

- set format

```
[set jnl_watch_time=journal-service-communication-response-wait-time]
```

Description

Operands in the set format

- `jnl_watch_time=`*journal-service-communication-response-wait-time*

  ~<unsigned integer>((0 to 65535))<<180>>(seconds)

  This operand is effective for the following:

  - `jnlopnfg` command
  - `jnlclsfg` command
  - `jnlchgfg` command
  - `jnlunlfg` command
  - `jnlswpfg` command#
  - Time to wait until a response to a swap request is received in the event of a journal file failure or full journal file
  - Time to wait until a response to completion of journal file I/O processing is received

  #

  The communication response wait time for the `jnlswpfg` command is twice the value of this operand.

  There are no other differences.

Client service definition

For details about the operands that are not explained here, see the list in *5.1(24) Client service definition*.

Formats

- set format

```
[set message_store_buflen=message-storage-buffer-pool-size]
```

Description

Operands in the set format

- `message_store_buflen=`*message-storage-buffer-pool-size*

  ~<unsigned integer>((1024 to 31457280))<<8196>>(bytes)

  If you change this operand's value while OpenTP1 is online, the `KFCA00850-E` message might be issued.

  The reason for this is that a shortage has occurred in the internal resources that had been allocated based on the operand value in effect before the change was made. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

  Note that when you change this operand's value, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

RAP-processing listener service definition

For details about the operands that are not explained here, see the list in *5.1(32) RAP-processing listener service definition*.

Formats

- set format

```
[set rap_parallel_server=RAP-processing-servers-count]
[set nice=change-of-process-priority]
[set
rap_listen_backlog=maximum-number-of-requests-that-can-be-placed-in-connection-queue]
```

Description

Operands in the set format

- `rap_parallel_server=`*RAP-processing-servers-count*

  ~<unsigned integer>((1 to 1024))<<1>>

  If you change this operand's value while OpenTP1 is online, the `KFCA00850-E` message might be issued.

  The reasons for this is that a shortage has occurred in the internal resources that had been allocated based on the operand value in effect before the change was made. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

  Note that when you change this operand's value, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

- `nice=`*change-of-process-priority*

  ~<unsigned integer>((0 to 39))<<0>>

  The meaning of this value varies. In OpenTP1 for Windows, this operand's value is interpreted as follows:

nice operand value	Priority to be applied
0 to 9	THREAD_PRIORITY_NORMAL
10 to 19	THREAD_PRIORITY_BELLOW_NORMAL
20 to 29	THREAD_PRIORITY_LOWEST
30 to 39	THREAD_PRIORITY_IDLE

- `rap_listen_backlog=`*maximum-number-of-requests-that-can-be-placed-in-connection-queue*

  ~<unsigned integer>((5 to OS's default value))<<OS's default value>>

  The specification range and default value vary. The OS's default value depends on the OS being used. For details about the OS's default value (real value), see the Release Notes. There are no other differences.

User service default definition

For details about the operands that are not explained here, see the list in *5.1(41) User service default definition*.

Formats

- set format

```
[set nice=change-of-process-priority]
[set message_store_buflen=message-storage-buffer-pool-size]
[set message_cell_size=scheduled-message-storage-cell-size]
[set prc_abort_signal=signal-number-for-abort]
[set process_privilege_restrict=Y|N]
[set process_privilege_name="privilege-name"[,"privilege-name"...]]
```

- putenv format

{{[putenv *environment-variable-name environment-variable-value*]}}

- dcputenv format

{{[dcputenv *environment-variable-name environment-variable-value*]}}

Description

Operands in the set format

- nice=*change-of-process-priority*

  ~<unsigned integer>((0 to 39))<<0>>

  The meaning of the value varies. In OpenTP1 for Windows, this operand's value is interpreted as follows:

nice operand value	Priority to be applied
0 to 9	THREAD_PRIORITY_NORMAL
10 to 19	THREAD_PRIORITY_BELLOW_NORMAL
20 to 29	THREAD_PRIORITY_LOWEST
30 to 39	THREAD_PRIORITY_IDLE

- message_store_buflen=*message-storage-buffer-pool-size*

  ~<unsigned integer>((1024 to 31457280))<<4096>>(bytes)

  If you change this operand's value while OpenTP1 is online, the KFCA00850-E message might be issued.

  The reason for this is that a shortage has occurred in the internal resources that had been allocated based on the operand value in effect before the change was made. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

  Note that when you change this operand's value, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

- message_cell_size=*scheduled-message-storage-cell-size*

  ~<unsigned integer>((512 to 31457280))<<512>>(bytes)

  If you change this operand's value while OpenTP1 is online, the KFCA00850-E message might be issued.

  The reason for this is that a shortage has occurred in the internal resources that had been allocated based on the operand value in effect before the change was made. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

  Note that when you change this operand's value, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

- prc_abort_signal=*signal-number-for-abort*

  ~<unsigned integer>((3, 6, 15))<<3>>

  Specify 3, 6, or 15 as the signal number. There are no other differences.

- process_privilege_restrict=Y|N

  ~<<N>>

  Specifies a default value for the process_privilege_restrict operand in the user service definition.

- process_privilege_name="*privilege-name*"[,"*privilege-name*"...]

  ~<character string>

  Specifies a default value for the process_privilege_name operand in the user service definition.

Operands in the putenv format

- *environment-variable-name environment-variable-value*

  ~<character string>

The differences from OpenTP1 for UNIX are as follows (there are no other differences):

- If you specify the `LANG` environment variable in the `putenv` format in the user service definition or user service default definition, specify the same value as in the system common definition. If this environment variable is not defined in the system common definition, do not specify any value other than `C`.
- If you specify the `PATH` environment variable in the `putenv` format in the user service definition or user service default definition, include in the specification the location of OpenTP1 commands, UAP load modules, and DLLs used by the UAP.

Operands in the dcputenv format

■ *environment-variable-name environment-variable-value*

~<character string>

Enclose each environment variable in percent signs (`%`), not dollar signs (`$`). There are no other differences.

User service definition

For details about the operands that are not explained here, see the list in *5.1(42) User service definition*.

Formats

- set format

```
  set module="executable-program-name"
[set nice=change-of-process-priority]
[set message_store_buflen=message-storage-buffer-pool-size]
[set message_cell_size=scheduled-message-storage-cell-size]
[set prc_abort_signal=signal-number-for-abort]
[set process_privilege_restrict=Y|N]
[set process_privilege_name="privilege-name"[,"privilege-name"...]]
```

- putenv format

{{[putenv *environment-variable-name environment-variable-value*]}}

- dcputenv format

{{[dcputenv *environment-variable-name environment-variable-value*]}}

Description

Operands in the set format

- module="*executable-program-name*"

  ~<identifier consisting of 1 to 14 characters>

  Do not specify the .exe extension in *executable-program-name*. For example, if the executable program name is spp.exe, specify set module ="spp". There are no other differences.

- nice=*change-of-process-priority*

  ~<unsigned integer>((0 to 39))

  The meaning of this value varies. In OpenTP1 for Windows, this operand value is interpreted as follows:

nice operand value	Priority to be applied
0 to 9	THREAD_PRIORITY_NORMAL
10 to 19	THREAD_PRIORITY_BELLOW_NORMAL
20 to 29	THREAD_PRIORITY_LOWEST
30 to 39	THREAD_PRIORITY_IDLE

- message_store_buflen=*message-storage-buffer-pool-size*

  ~<unsigned integer>((1024 to 31457280))(bytes)

  If you change this operand's value while OpenTP1 is online, the KFCA00850-E message might be issued.

  The reason for this is that a shortage has occurred in the internal resources that had been allocated based on the operand value in effect before the change was made. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

  Note that when you change this operand's value, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

- message_cell_size=*scheduled-message-storage-cell-size*

  ~<unsigned integer>((512 to 31457280))(bytes)

  If you change this operand's value while OpenTP1 is online, the KFCA00850-E message might be issued.

  The reason for this is that a shortage has occurred in the internal resources that had been allocated based on the operand value in effect before the change was made. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

  Note that when you change this operand's value, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

- prc_abort_signal=*signal-number-for-abort*

  ~<unsigned integer>((3, 6, 15))

  Specify 3, 6, or 15 as the signal number. There are no other differences.

- process_privilege_restrict=Y|N

  ~<<N>>

  Specifies whether the user server's Windows privileges are to be limited.

  Y

Limit the user server's privileges (other than SeChangeNotifyPrivilege).

When you specify Y, you can grant privileges to the user server by specifying the privileges to be granted in the process_privilege_name operand.

N

The user server is to have all the privileges set for the service logon account.

- process_privilege_name=*privilege-name*"[,"*privilege-name*"...]

  ~<character string>

  Specifies the names of Windows privileges (such as SeSecurityPrivilege) that are to be granted to the user server.

  This operand is effective only when Y is specified in the process_privilege_restrict operand.

  A privilege specified in this operand will not be granted to the user server if that privilege is not set for the service logon account. When you grant a privilege, evaluate whether the privilege is really needed.

putenv format

- *environment-variable-name environment-variable-value*

  ~<character string>

  The differences from OpenTP1 for UNIX are as follows (there are no other differences):

  - If you specify the LANG environment variable in the putenv format in the user service definition or user service default definition, specify the same value as in the system common definition. If this environment variable is not defined in the system common definition, do not specify any value other than C.
  - If you specify the PATH environment variable in the putenv format in the user service definition or user service default definition, include in the specification the location of OpenTP1 commands, UAP load modules, and DLLs used by the UAP.

dcputenv format

- *environment-variable-name environment-variable-value*

  ~<character string>

  Enclose each environment variable in percent signs (%), not dollar signs ($). There are no other differences.

Notes

If you add a new user service definition while OpenTP1 is online, the KFCA00850-E

message might be issued.

The reason for this is that a shortage has occurred in the internal resources that had been allocated based on the definition in effect before the new definition was added. You can prevent this from occurring by terminating OpenTP1 and then restarting it.

Note that if you add a user service definition, the required size of the static shared memory changes. For the formula for determining the memory size, see the manual *OpenTP1 Version 7 System Definition*.

Definition of system service information

For details about the operands that are not explained here, see the list in *5.1(47) Definition of system service information*.

Formats

- set format

```
set module="executable-program-name"
```

Description

Operands in the set format

- `module="`*executable-program-name*`"`

  ~<identifier consisting of 1 to 8 characters>

  Do not specify the `.exe` extension in *executable-program-name*. For example, if the executable program name is `mcfu01.exe`, specify `set module ="mcfu01"`. There are no other differences.

Chapter

6. Operations

The operation methods differ between OpenTP1 for Windows and OpenTP1 for UNIX. OpenTP1 for Windows provides a graphical user interface (GUI) for some operations. This chapter explains the operations that are specific to OpenTP1 for Windows, as well as the GUI provided by OpenTP1 for Windows.

For information common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the OpenTP1 series manuals.

6.1 Overview of OpenTP1 for Windows operations
6.2 Starting OpenTP1
6.3 Terminating OpenTP1
6.4 MultiOpenTP1 operations
6.5 System operations using scenario templates
6.6 Troubleshooting
6.7 GUI provided by OpenTP1 for Windows

6.1 Overview of OpenTP1 for Windows operations

A major difference between OpenTP1 for Windows and OpenTP1 for UNIX is that OpenTP1 for Windows runs as a Windows service. The installer (the `dcsetupml` command for a multiOpenTP1) registers services into and deletes services from Windows. The service name is **OpenTP1** (**OpenTP1**_*XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier).

There are also differences in the system operating methods between OpenTP1 for Windows and OpenTP1 for UNIX. This chapter explains system operations specific to OpenTP1 for Windows. For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

The following table provides an overview of the OpenTP1 for Windows operations explained in this chapter.

Table 6-1: Overview of OpenTP1 for Windows operations

Operation	Overview	Section
Starting OpenTP1	Explains how to start OpenTP1. In OpenTP1 for Windows, three startup methods are provided: • From the Windows Services dialog box • With a command • With the GUI	6.2
Terminating OpenTP1	Explains how to terminate OpenTP1. In OpenTP1 for Windows, three termination methods are provided: • From the Windows Services dialog box • With a command • With the GUI	6.3
Running a multiOpenTP1	Explains how to run a multiOpenTP1 in OpenTP1 for Windows. The command input environment and operation commands used in OpenTP1 for Windows differ from those in OpenTP1 for UNIX.	6.4
System operation using scenario templates	Explains how to use scenario templates to run the system in OpenTP1 for Windows.	6.5
Troubleshooting	Explains troubleshooting in OpenTP1 for Windows.	6.6
Graphical user interface (GUI) in OpenTP1 for Windows	Explains the GUI provided in OpenTP1 for Windows.	6.7

6.2 Starting OpenTP1

Three methods are provided for starting OpenTP1 for Windows:

- From the Windows Services dialog box
- With a command
- With the GUI

This section explains the three startup methods. It also explains how the startup mode is determined in OpenTP1 for Windows.

6.2.1 Starting OpenTP1 from the Windows Services dialog box

In the Windows Services dialog box, you can start OpenTP1 in the following modes:

- Manual start
- Automatic start
- Forced normal start

This subsection explains these three startup modes.

Reference note:

You open the Windows Services dialog box from **Control Panel**.

(1) Manual start of OpenTP1

To start OpenTP1 manually:

1. Open the Windows Services dialog box.
2. From the displayed list of services, select **OpenTP1** (or **OpenTP1_***XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier).
3. Click the **Start** button.

   The Service Control dialog box is displayed and OpenTP1 starts.
4. Click the **OK** button.

(2) Automatic start of OpenTP1

In this mode, OpenTP1 starts automatically when Windows starts.

To set OpenTP1 to start automatically:

1. Open the Windows Services dialog box.
2. In the displayed list of services, double-click **OpenTP1** (or **OpenTP1_***XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier).

3. Change **Startup type** to **Automatic**.

(3) Forced normal start of OpenTP1

To start OpenTP1 in the forced normal startup mode:

1. Open the Windows Services dialog box.
2. In the displayed list of services, double-click **OpenTP1** (or **OpenTP1**_*XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier).
3. In the **Start parameters** text box, enter `normal`.
4. Click the **Start** button.

(4) Notes

Always start OpenTP1 with the **Allow service to interact with desktop** check box selected (the **Allow service to interact with desktop** check box is located on the **Log On** page in the OpenTP1 Properties dialog box). If this check box is not selected, the system might shut down due to a shortage of desktop heap. To display the OpenTP1 Properties dialog box, in the OpenTP1 in the Windows Services dialog box, right-click the service name of OpenTP1 and then choose **Properties**.

If you start OpenTP1 in this status, the `KFCA26524-W` message will be output to the Event Viewer. Even though this message is output, OpenTP1 starts normally. However, we recommend that you terminate OpenTP1, go to the **Log On** page and select the **Allow service to interact with desktop** check box, and then restart OpenTP1.

In OpenTP1 Version 6 or later, the console is not displayed by default. To display the console, use the OpenTP1 console output facility. For details about the OpenTP1 console output facility, see *2.2.3 OpenTP1 console output facility*.

6.2.2 Starting OpenTP1 with a command

You can start OpenTP1 by entering one of the following commands from the command prompt:

- `net start`

  Specify `OpenTP1` in the argument (or `OpenTP1`_*XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier).

- `ntbstart`

  For a multiOpenTP1, you must execute this command from the command prompt at the console that was created by `dcmakecon` *XXXX* (where *XXXX* is a user-specified identifier).

In OpenTP1 for Windows, OpenTP1 cannot be started with the `dcstart` command.

To start OpenTP1 in the forced normal startup mode from the command prompt,

execute the `ntbstart -n` command. For a multiOpenTP1, you must execute this command from the command prompt at the console that was created by `dcmakecon` *XXXX* (where *XXXX* is a user-specified identifier).

For details about the `ntbstart` command, see *ntbstart (starts OpenTP1)* in Chapter *7. Operation Commands*. For details about the `dcmakecon` command, see *dcmakecon (creates a multiOpenTP1 console)* in Chapter *7. Operation Commands*. For details about the `net start` command, see the Windows documentation.

6.2.3 Starting OpenTP1 with the GUI

You can use the GUI provided by OpenTP1 for Windows to start OpenTP1.

For details about the GUI provided by OpenTP1 for Windows, see *6.7 GUI provided by OpenTP1 for Windows*.

6.2.4 Determining the startup mode

The combination of the startup method and the startup mode is called the *startup mode*.

The startup mode is determined by the previous termination mode and the value of the `mode_conf` operand in the system environment definition. In OpenTP1 for Windows, you can specify only `AUTO` in the `mode_conf` operand in the system environment definition (`MANUAL1` and `MANUAL2` are not supported). Therefore, the conditions for determining the startup mode in OpenTP1 for Windows differ from those in OpenTP1 for UNIX.

The following table shows the startup modes in OpenTP1 for Windows.

Table 6-2: Startup modes in OpenTP1 for Windows

Conditions for determining startup mode		Startup mode	
Previous termination mode	**mode_conf value**	**Startup method**	**Startup mode**
Normal termination	`AUTO`	Manual#1	Normal start
Forced normal termination	`AUTO`	Manual#1	Normal start
Planned termination A	`AUTO`	Manual#1	Restart#2
Planned termination B	`AUTO`	Manual#1	Restart#2
Forced termination	`AUTO`	Manual#1	Restart#2
Abnormal termination	`AUTO`	Automatic	Restart

#1

OpenTP1 starts automatically when Windows starts.

#2

You can also start OpenTP1 in the forced normal startup mode from the Windows Services dialog box, a command, or the GUI.

Because OpenTP1 for Windows does not support the `dcstart` command, the `dcstart -n` command cannot be used to start OpenTP1 in the forced normal startup mode.

6.3 Terminating OpenTP1

Three methods are provided for terminating OpenTP1 for Windows:

- From the Windows Services dialog box
- With a command
- With the GUI

This section explains the three termination methods.

6.3.1 Terminating OpenTP1 from the Windows Services dialog box

To terminate OpenTP1 from the Windows Services dialog box:

1. Open the Windows Services dialog box.
2. From the displayed list of services, select **OpenTP1** (or **OpenTP1**_*XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier).
3. Click the **Stop** button.

   OpenTP1 terminates in the normal termination mode.

6.3.2 Terminating OpenTP1 with a command

You can terminate OpenTP1 by entering one of the commands listed below from the command prompt. OpenTP1will terminate in the normal termination mode.

- `net stop`

  Specify `OpenTP1` in the argument (or `OpenTP1_`*XXXX* for a multiOpenTP1, where *XXXX* is a user-specified identifier). If the command cannot terminate OpenTP1 normally for some reason, OpenTP1 is terminated with a forced normal termination. If forced normal termination also fails, OpenTP1 is terminated forcibly. Note that once OpenTP1 has been terminated forcibly, it must be restarted (rerun) to start OpenTP1 operation.

- `dcstop`

  For a multiOpenTP1, execute this command from the command prompt at the console that was created by `dcmakecon` *XXXX* (where *XXXX* is a user-specified identifier).

For details about the `dcstop` command, see *List of operation commands* in Chapter *7. Operation Commands*. For details about the `dcmakecon` command, see *dcmakecon (creates a multiOpenTP1 console)* in Chapter *7. Operation Commands*. For details about the `net stop` command, see the Windows documentation.

6.3.3 Terminating OpenTP1 with the GUI

You can use the GUI provided by OpenTP1 for Windows to terminate OpenTP1.

For details about the GUI provided by OpenTP1 for Windows, see *6.7 GUI provided by OpenTP1 for Windows*.

6.3.4 Notes about OpenTP1 termination

If a system failure occurs in OpenTP1 and then OpenTP1 terminates during system restart processing that was initiated using either of the following methods, the OpenTP1 service status might remain as `Stopping`:

- Termination of OpenTP1 from the Windows Services dialog box
- Termination of OpenTP1 with a command

If such a failure occurs, first check the cause of the failure and take appropriate action. Then execute the `dcstop` command to terminate OpenTP1. For a multiOpenTP1, execute the `dcstop` command at the console that was created with the `dcmakecon` command. For details about the `dcstop` command, see *List of operation commands* in Chapter *7. Operation Commands*. For details about the `dcmakecon` command, see *dcmakecon (creates a multiOpenTP1 console)* in Chapter *7. Operation Commands*.

6.4 MultiOpenTP1 operations

This section explains the multiOpenTP1 operations in OpenTP1 for Windows that differ from OpenTP1 for UNIX. The differences are found in the command input environment and in operation commands.

For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

(1) Command input environment

The command input environment differs between OpenTP1 for Windows and OpenTP1 for UNIX as follows:

- In the environment for the original OpenTP1

  You enter OpenTP1 commands from the command prompt.

  When you manipulate the original OpenTP1, you can execute OpenTP1 commands not only from the command prompt but also from the console created with the `dcmakecon` command.

- In a multiOpenTP1 environment

  You must execute the `dcmakecon` command to create the console that is to be used to run the specific OpenTP1. You use this console to execute commands in the environment for the multiOpenTP1.

  When you manipulate a multiOpenTP1, you execute commands at the console created with the `dcmakecon` command. Note that you can also execute `net start OpenTP1_`*XXXX* and `net stop OpenTP1_`*XXXX* (where *XXXX* is a user-specified identifier) from the regular command prompt.

For details about starting OpenTP1, see *6.2 Starting OpenTP1*. For details about terminating OpenTP1, see *6.3 Terminating OpenTP1*.

(2) Commands for running a multiOpenTP1

You can use the following commands to run a multiOpenTP1:

- `dcdls`: Checks the setup status of the multiOpenTP1.
- `dcmakecon`: Creates the multiOpenTP1 console.
- `dcsetupml`: Sets up the multiOpenTP1 or cancels setup of the multiOpenTP1.

For details about these commands, see *Details of the commands specific to OpenTP1 for Windows* in Chapter *7. Operation Commands*.

6.5 System operations using scenario templates

This section explains system operations using scenario templates in OpenTP1 for Windows that differ from those in OpenTP1 for UNIX. For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the description of operations with JP1 linked in the manual *OpenTP1 Version 7 Operation*.

6.5.1 Registering scenarios

In the scenario registration process, the storage location of the *scenario template definition file* differs between OpenTP1 for Windows and OpenTP1 for UNIX. For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the description of scenario registration in the manual *OpenTP1 Version 7 Operation*.

The scenario template definition file (`TP1_ServerBase.sjis.xml`) is stored in the following folder:
OpenTP1-installation-folder`\jp1_template\ScenarioTemplate\`

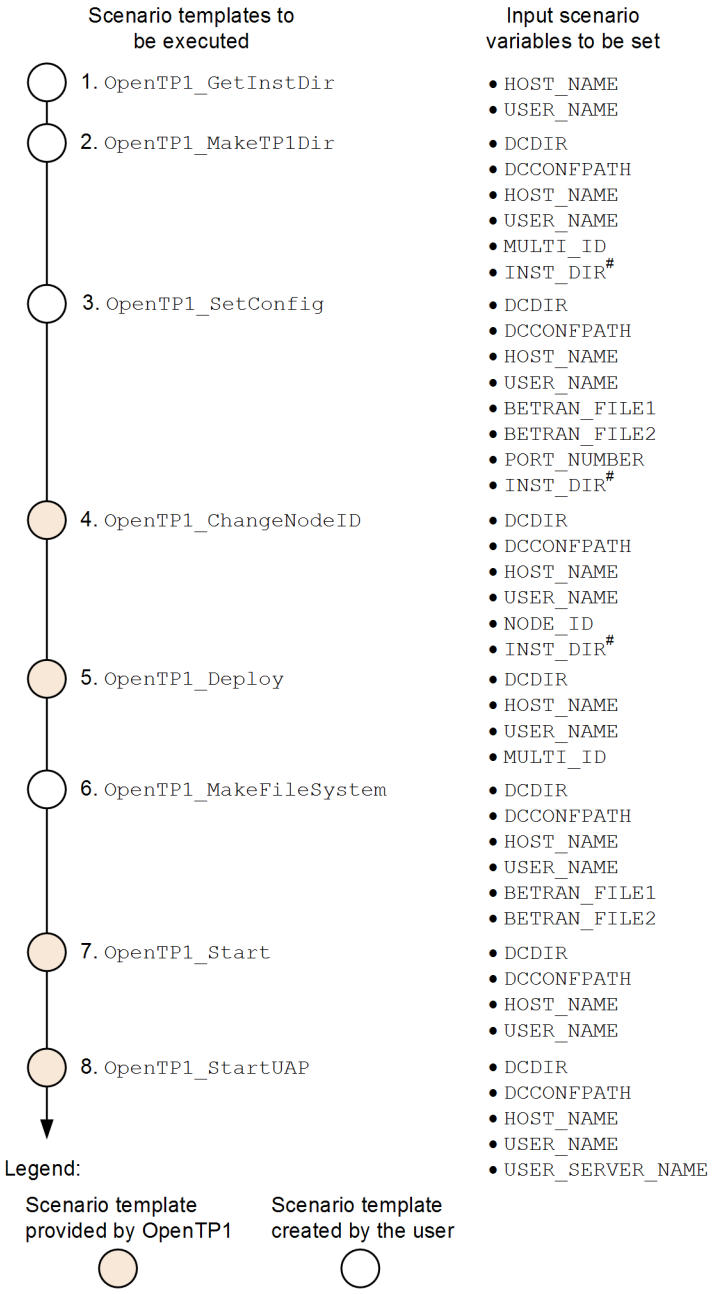
6.5.2 Scale-out operation

In the scale-out scenario configuration for scale-out operations, the scenario for adding a new OpenTP1 node differs between OpenTP1 for Windows and OpenTP1 for UNIX. For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the description of scale-out operations in the manual *OpenTP1 Version 7 Operation*.

(1) Scenario for adding a new OpenTP1 node

The following figure shows the procedure for executing the scenario for adding a new OpenTP1 node.

Figure 6-1: Procedure for executing the scenario for adding a new OpenTP1 node



#

INST_DIR is a variable that is inherited from OpenTP1_GetInstDir.

The table below lists and describes the scenario templates. The numbers in the table correspond to the numbers in the figure.

Table 6-3: Names and description of scenario templates

No.	Scenario template name	Description
1	OpenTP1_GetInstDir#1	Acquires the OpenTP1 installation folder
2	OpenTP1_MakeTP1Dir#1	Creates an OpenTP1 folder
3	OpenTP1_SetConfig#1	Specifies OpenTP1 definitions
4	OpenTP1_ChangeNodeID#2	Specifies a node ID
5	OpenTP1_Deploy#2	Registers OpenTP1
6	OpenTP1_MakeFileSystem#1	Creates an OpenTP1 file system
7	OpenTP1_Start#2	Starts OpenTP1
8	OpenTP1_StartUAP#2	Starts a UAP

#1

This scenario template is created by the user. For details about the sample scenario templates provided by OpenTP1, see the description of using the sample scenario templates in the manual *OpenTP1 Version 7 Operation*.

#2

This scenario template is provided by OpenTP1. For details, see the description of scenario templates in the manual *OpenTP1 Version 7 Operation*. For details about OpenTP1_Deploy, see *6.5.4(1) OpenTP1_Deploy*.

6.5.3 Using scenario templates

The scale-out scenario templates provided by OpenTP1 include sample templates for scenarios used to add a new node.

Some of the sample scenario templates related to multiOpenTP1 differ between OpenTP1 for Windows and OpenTP1 for UNIX. For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the description of using the sample scenario templates in the manual *OpenTP1 Version 7 Operation*.

One of the sample scenario templates for OpenTP1 for Windows (OpenTP1_ScenarioScaleout) configures a multiOpenTP1 environment or an

original OpenTP1 environment installed by the installer.

(1) Names and processing of sample scenario templates

The following table lists and describes the sample scenario templates.

Table 6-4: Names and processing of sample scenario templates

No.	Name of sample scenario template	Processing	Input scenario variables to be referenced
1	OpenTP1_GetInstDir	Acquires the OpenTP1 installation folder	HOST_NAME USER_NAME
2	OpenTP1_MakeTP1Dir	Creates an OpenTP1 folder	DCDIR DCCONFPATH HOST_NAME USER_NAME MULTI_ID INST_DIR#
3	OpenTP1_SetConfig	Specifies OpenTP1 definition settings	DCDIR DCCONFPATH HOST_NAME USER_NAME BETRAN_FILE1 BETRAN_FILE2 PORT_NUMBER INST_DIR#
4	OpenTP1_ChangeNodeID	Sets a node ID	DCDIR DCCONFPATH HOST_NAME USER_NAME NODE_ID INST_DIR#
5	OpenTP1_Deploy	Registers OpenTP1	DCDIR HOST_NAME USER_NAME MULTI_ID
6	OpenTP1_MakeFileSystem	Creates an OpenTP1 file system	DCDIR DCCONFPATH HOST_NAME USER_NAME BETRAN_FILE1 BETRAN_FILE2

No.	Name of sample scenario template	Processing	Input scenario variables to be referenced
7	OpenTP1_Start	Starts OpenTP1	DCDIR DCCONFPATH HOST_NAME USER_NAME
8	OpenTP1_StartUAP	Starts a UAP (sample program)	DCDIR DCCONFPATH HOST_NAME USER_NAME USER_SERVER_NAME

#

There is no need for the user to enter the INST_DIR input scenario variable because it is acquired by OpenTP1_GetInstDir and passed to the scenario templates used in the subsequent processing.

The details of the sample scenario templates are provided below, where the numbers correspond to the numbers in the table.

1. OpenTP1_GetInstDir (acquires the OpenTP1 installation folder)

   This sample scenario template acquires the DCDIR environment variable as the OpenTP1 installation folder. The acquired OpenTP1 installation folder is passed to the OUT_INST_DIR output scenario variable.

2. OpenTP1_MakeTP1Dir (creates an OpenTP1 folder)

   This sample scenario template creates the OpenTP1 folder specified in the DCDIR input scenario variable by using an OpenTP1 folder creation batch file (*OpenTP1-installation-folder*\jp1_template\examples\tools\dcjmk_dcdir.bat). This batch file processing is not performed for the original OpenTP1 because its folder will have already been created.

   In OpenTP1 for Windows, user, group, and mode settings are not specified.

3. OpenTP1_SetConfig (specifies OpenTP1 definition settings)

   This sample scenario template uses an OpenTP1 environment setup batch file (*OpenTP1-installation-folder*\jp1_template\examples\tools\dcjset_conf.bat) to configure the OpenTP1 system definitions and the load modules in the following order:

   - Creates the following folders:

     For a multiOpenTP1: %DCDIR%\aplib, %DCDIR%\jp1_template\tools, and %DCCONFPATH%

For the original OpenTP1: %DCDIR%\jp1_template\tools

- Copies aplib, conf, and tools from the *OpenTP1-installation-folder*\jp1_template\examples folder to the created folder.
- Specifies the node-specific information (such as the file system path name) in the system definitions.

4. OpenTP1_ChangeNodeID (sets a node ID)

   This sample scenario template sets the node ID of OpenTP1 by using the value of the NODE_ID input scenario variable.

5. OpenTP1_Deploy (registers OpenTP1)

   For a multiOpenTP1, this sample scenario template executes the dcsetupml command to set up the multiOpenTP1.

   This processing is not performed for the original OpenTP1, because OpenTP1 will have already been registered.

6. OpenTP1_MakeFileSystem (creates an OpenTP1 file system)

   This sample scenario template uses an OpenTP1 file creation batch file (%DCDIR%\jp1_template\examples\tools\dcj_mkfs.bat) to create an OpenTP1 file system and OpenTP1 files. The name of the OpenTP1file system is the values of the BETRAN_FILE1 and BETRAN_FILE2 input scenario variables.

7. OpenTP1_Start (starts OpenTP1)

   This sample scenario template starts OpenTP1.

8. OpenTP1_StartUAP (starts a UAP (sample program))

   This sample scenario template starts a UAP of the user server set in the USER_SERVER_NAME input scenario variable.

(2) Registering sample scenario templates

Before you can execute the sample scenario templates, you must use JP1/AJS2 - Scenario Operation View to register them into JP1/AJS2 - Manager.

The sample scenario templates are stored at the following location:
/ScenarioLibrary/OpenTP1/TP1_ServerBase/
OpenTP1_ScenarioScaleout

In order to register the sample scenario templates, you must set the input scenario variables. The following table lists and describes the information to be set in the sample scenario templates.

Table 6-5: Information to be set in the sample scenario templates

Input scenario variable	Description
DCDIR	Home folder of OpenTP1
DCCONFPATH	Folder that stores the OpenTP1 definition files
HOST_NAME#1	Host name of the OpenTP1 node that is to execute the scenario job
USER_NAME#1	OpenTP1 user name
BETRAN_FILE1#2	Path of the OpenTP1 file system
BETRAN_FILE2#2	Path of the OpenTP1 file system
USER_SERVER_NAME	User server name
NODE_ID	OpenTP1 node identifier (unique string of 4 characters assigned to each node)
PORT_NUMBER	OpenTP1's name port number
MULTI_ID#3	MultiOpenTP1 identifier (string of 4 characters that is unique within the node)

#1

This setting can be omitted.

#2

If you define a character special file, you must allocate a partition before you execute a scenario.

#3

Omit this setting for the original OpenTP1.

6.5.4 Details of scenario templates

Specifications of the following scenario templates differ between OpenTP1 for Windows and OpenTP1 for UNIX:

- OpenTP1_Deploy#
- OpenTP1_Undeploy#
- OpenTP1_ScenarioScaleout

#

Not executed for the original OpenTP1.

This subsection provides the details of these three scenario templates. For details about

the other scenario templates, see the descriptions providing the details of the scenario templates in the manual *OpenTP1 Version 7 Operation*.

(1) OpenTP1_Deploy

This scenario template sets up a multiOpenTP1 in a specified OpenTP1 folder. If the specified OpenTP1 folder is the original OpenTP1, this scenario template does not execute.

Input scenario variables

- `DCDIR` ~<string of 1 to 50 characters>

  Specifies the OpenTP1 folder name to be set up.

  This value must be unique within the OpenTP1 system.

- `HOST_NAME`

  Specifies the host name of the OpenTP1 node that is to execute the scenario job.

  You can omit this input scenario variable. If it is omitted, the scenario job is executed on the target host of the scenario.

- `USER_NAME`

  Specifies the user name of the OpenTP1 administrator that is to execute the scenario job.

  You can omit this input scenario variable. If it is omitted, the primary user in the user mapping defined in JP1/Base executes the scenario job.

- `MULTI_ID` ~<string of 1 to 4 characters>

  Specifies the identifier of a multiOpenTP1.

  You can omit this input scenario variable if the OpenTP1 folder specified in the `DCDIR` input scenario variable is the original OpenTP1.

Executing user

OpenTP1 administrator specified in the `USER_NAME` input scenario variable

(2) OpenTP1_Undeploy

This scenario template deletes a multiOpenTP1 when the specified OpenTP1 folder is a multiOpenTP1. If the specified OpenTP1 folder is the original OpenTP1, this scenario template does not execute.

Input scenario variables

- `DCDIR` ~<string of 1 to 50 characters>

  Specifies the name of the OpenTP1 folder to be deleted.

- HOST_NAME

  Specifies the host name of the OpenTP1 node that is to execute the scenario job.

  You can omit this input scenario variable. If it is omitted, the scenario job is executed on the target host of the scenario.

- USER_NAME

  Specifies the user name of the OpenTP1 administrator that is to execute the scenario job.

  You can omit this input scenario variable. If it is omitted, the primary user in the user mapping defined in JP1/Base executes the scenario job.

Executing user

OpenTP1 administrator specified in the USER_NAME input scenario variable

(3) OpenTP1_ScenarioScaleout

This scenario template sets up an OpenTP1 environment and then starts OpenTP1 and the sample SPP.

Input scenario variables

- DCDIR ~<string of 1 to 50 characters>

  Specifies a name for the OpenTP1 folder to be set up.

  This value must be unique within the OpenTP1 system.

- DCCONFPATH

  Specifies the folder that stores the OpenTP1 definition files.

- HOST_NAME

  Specifies the host name of the OpenTP1 node that is to execute the scenario job.

  You can omit this input scenario variable. If it is omitted, the scenario job is executed on the target host of the scenario.

- USER_NAME

  Specifies the user name of the OpenTP1 administrator that is to execute the scenario job.

  You can omit this input scenario variable. If it is omitted, the primary user in user mapping defined in JP1/Base executes the scenario job.

- BETRAN_FILE1

Specifies the path of the OpenTP1 file system for the primary system.

If you specify a character special file, you must allocate a partition before you execute the scenario template.

- BETRAN_FILE2

  Specifies the path of the OpenTP1 file system for the secondary system.

  If you specify a character special file, you must allocate a partition before you execute the scenario template.

- USER_SERVER_NAME ~<string of 1 to 8 characters>

  Specifies the name of the user server to be started.

  If you use sample scenario templates, specify basespp.

- NODE_ID ~<string of 4 characters>

  Specifies the node ID of the OpenTP1 to be set up.

  This value must be unique within the OpenTP1 system.

- PORT_NUMBER ~((5001 to 65535))

  Specifies the port number to be used by the name server at the new OpenTP1 node being added to the OpenTP1 system.

- MULTI_ID ~<string of 1 to 4 characters>

  Specifies the identifier of a multiOpenTP1.

  You can omit this input scenario variable if the OpenTP1 folder specified in the DCDIR input scenario variable is the original OpenTP1.

Executing user

OpenTP1 administrator specified in the USER_NAME input scenario variable

6.6 Troubleshooting

Troubleshooting is described in the manual *OpenTP1 Version 7 Operation*. This section explains those elements of troubleshooting that are specific to OpenTP1 for Windows.

6.6.1 Error events and causes

The table below lists the error events specific to OpenTP1 for Windows, their possible causes, and the actions to be taken. For details about other error events, causes, and actions to be taken, see the manual *OpenTP1 Version 7 Operation*.

Table 6-6: Error events, possible causes, and actions to be taken

Event	Cause	Action
UAP will not start.	Process-specific memory is insufficient.	If there is an unneeded process, terminate it. If there are no unneeded processes, add physical memory and extend the Windows virtual memory page file.
UAP will not start (the user server shuts down with termination status `0x00000080` or `0xC0000135`).	The path for the runtime library required for UAP execution has not been set.	Set the path for the runtime library required for UAP execution. For details, see *4.3.3 Setting the runtime library path*.
	A manifest file is missing, or the library specified in the manifest file cannot be referenced.	Check the manifest file or library. For details, see *3.2.1(1) Preparing a manifest file* or *3.3(4) Notes about using Visual Studio*.
UAP cannot be terminated.	An application exception has occurred.	For the action to be taken, see *6.6.2 Action to be taken in the event of an application exception*.
OpenTP1 will not start.	Memory or disk capacity is insufficient.	Check the OpenTP1 definitions, and revise them as necessary. If OpenTP1 still will not start, take appropriate action for the following causes: • When the process-specific memory is insufficient Check the sizes of physical memory and the Windows virtual memory page file, and revise them as necessary. • When a shared memory is insufficient Delete any unneeded files so that the available capacity in the disk partition in which OpenTP1 is installed is greater than the size of the shared memory allocated by OpenTP1.

Event	Cause	Action
Communication performance or disk I/O performance is poor.	An anti-virus program that quarantines executable programs in real-time might be running on the computer. Depending on the anti-virus program, OpenTP1 might be temporarily unable to access a file required for operation, thereby adversely affecting OpenTP1 operation.	We recommend that you remove the following paths as targets for quarantining by the anti-virus program: • Folder paths under `%DCDIR%` • If the definition files and the TP1 file system are stored under folders other than `%DCDIR%`, those folders' paths
There are not enough ports.	Traffic is high.	For the action to be taken, see *6.6.3 Action to be taken in the event of a shortage of ports (tuning the number of TCP/IP ports)*.

6.6.2 Action to be taken in the event of an application exception

If an application exception occurs in a process running under OpenTP1 during online operation, the Windows debug handler is called and the debug program set in the registry is started. If the debug program to be started is an interactive program, such as VC++, the OpenTP1 system might be unable to restart (rerun) because the program resulting in the exception cannot be terminated. In Windows, if the system environment itself requires recovery processing, configure the system so that Dr. Watson is specified in the registry.

Note that you can set Dr. Watson only if you use Windows XP or earlier or Windows Server 2003 or earlier. If you use any other edition of Windows, use Windows Error Reporting (WER) to specify the settings so that a crush dump is acquired.

(1) Specifying output environment settings for Dr. Watson

For details about Dr. Watson, see Windows Help.

To specify environment settings to enable Dr. Watson:

1. Make a registry backup.

   Start the registry editor (`regedit.exe`), open the `\\HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\AeDebug` key, and then make a backup of the `Debugger` and `Auto` entry values. After that, terminate the registry editor.

2. From the command prompt, execute the following command to enable the Dr. Watson setting:

```
drwtsn32 -i
```

3. Make note of all settings (for example, by writing down the settings displayed in the Dr. Watson dialog box).

4. Specify options for the Dr. Watson settings, as necessary.

Specify the following settings:

- In **Log File Path**, specify a valid folder (you can also use the default setting).
- In **Crash Dump**, specify a valid path (you can also use the default setting).
- In **Crash Dump Type**, select **Full** (applicable only in Windows Server 2003).
- Select the following items in the **Options** check boxes:

  Dump Symbol Table

  Dump All Thread Contexts

  Create Crash Dump File

These option settings are not mandatory, but we recommend that you create a crash dump file because it is useful for troubleshooting in the event of an application exception.

5. Click the **OK** button to exit Dr. Watson.

(2) Restoring debug program information from a backup

To restore the debug program information from a backup:

1. Restore the debug program information from the backup registry.

   Start the registry editor (`regedit.exe`), open the `\\HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\AeDebug` key, and then set the `Debugger` and `Auto` entries to the values used in the backup. After that, terminate the registry editor.

2. To restore the option settings for Dr. Watson, specify the values used in the backup.

3. Click the **OK** button to exit Dr. Watson.

6.6.3 Action to be taken in the event of a shortage of ports (tuning the number of TCP/IP ports)

If you configure a large-scale system that uses OpenTP1, a shortage of ports managed by TCP/IP might occur. In such a case, you can resolve the shortage by performing temporary close processing.

However in a high-traffic environment, a shortage of ports might not be resolved by performing temporary close processing. In such a case, you can tune the ports used on the corresponding platform by extending ephemeral ports in Windows settings.

Use a Windows registry command to set the following parameters:

- `MaxUserPort`

- TcpTimedWaitDelay
- ReservedPorts

For details about setting these parameters, see the Windows documentation.

For details about temporary close processing, see the description of how to limit the number of ports in the manual *OpenTP1 Version 7 Description*.

6.6.4 Using the information acquired in the event of a failure

This subsection explains how to use acquired information in order to determine the cause of a failure.

For details about the information to be acquired in the event of a failure, see the descriptions of the files that are output by OpenTP1 in the manual *OpenTP1 Version 7 Operation*.

(1) Using standard output and standard error

OpenTP1 for Windows enables you to output to a file and to the console the contents of standard output and standard error by the system server and user server (this includes KFCA-*xxxxx* messages and all information specified for standard output by the user).

For details about the functions for checking standard output and standard error, see *2.2 Functions for checking standard output and standard error*.

(2) Checking event logs in Windows Event Viewer

The following information is output to the Windows **Event Viewer**:

- OpenTP1 messages (other than messages output by the installation and setup programs and operation commands)
- Messages whose output destination is indicated in the manuals *OpenTP1 Version 7 Messages* as being standard output or standard error.

To view OpenTP1's event logs:

1. Display the Windows **Event Viewer**.
2. Choose **Applications**.

   The application logs are displayed. An event for which **OpenTP1** is displayed under **Source** is a message issued by OpenTP1.

The messages specified in the dc_logprint function or CBLDCLOG('PRINT ') are also displayed in the Windows **Event Viewer**. Note the following about using the dc_logprint function or CBLDCLOG('PRINT '):

- CBLDCLOG('PRINT ') ignores specification of message colors.
- If an invalid argument value is set, an error message is output to the event log in

the following format:

```
(xxx...xx) cannot be output due to an invalid message
```

Legend:

xxx...xx: Message ID

- In the message format `KFCAn1n2n3n4n5-x`, only `-I`, `-R`, `-W`, or `-E` can be specified for `-x`. If any other value is specified, the specified argument is treated as being invalid and the error message shown above is issued. Also, `-x` corresponds to the **Type** in the Windows **Event Viewer**. The following shows the type of information that is displayed for each `-x` value:
  - `-I` or `-R`: Informational
  - `-W`: Warning
  - `-E`: Error

6.7 GUI provided by OpenTP1 for Windows

OpenTP1 for Windows enables you to use a graphical user interface (GUI) to set up an environment for OpenTP1 and to start and terminate OpenTP1. Because these operations are independent of each other, you can, for example, use the GUI to set up an environment and then enter commands at the command prompt to start OpenTP1.

Check the following before you use the GUI provided by OpenTP1 for Windows:

- .NET Framework version

  .NET Framework 2.0 or later must be installed in order to use the GUI. Install .NET Framework 2.0 or later, if necessary.

- User account control settings

  If you use the GUI in Windows Vista or later, or Windows Server 2008 or later, the User Account Control dialog box is displayed before you start the GUI. Click the **Allow** button.

- Security policy settings

  Specify full trust in the files listed below. The GUI cannot be used without the full trust setting. For details about how to specify the security policy settings, see the .NET Framework documentation.

  - `%DCDIR%\tp1_tools\Control.exe`
  - `%DCDIR%\tp1_tools\Environment.exe`

- Output destination of command logs and audit logs

  Command logs and audit logs might be output when you use the GUI. However, these logs are not output to a multiOpenTP1 regardless of the target of the GUI operations.

6.7.1 GUI for environment setup

This subsection explains use of the GUI to set up an OpenTP1 environment. The GUI provides as templates the minimum system definitions required in order to use OpenTP1. The provided templates include multiple models in which appropriate values (parameters) are set for various system configurations. You can use the GUI to change the set parameter values. Parameter values specified by using the GUI are treated as operand values in the corresponding system definitions.

Note:

OpenTP1 system definitions that are not included in the provided templates cannot be added to those templates.

To use the GUI to set up an OpenTP1 environment:

1. From the **Start** menu, choose **Programs**, **OpenTP1**, and then **OpenTP1 Environment**.

   The OpenTP1 Environment window is displayed.

2. In the OpenTP1 Environment window, select a desired definition type and model, and then click the **Next** button.

   The OpenTP1 Environment window (parameter information) is displayed.

3. Enter values for parameters in the OpenTP1 Environment window (parameter information).

4. To check the specified settings, click the **Details** button in the OpenTP1 Environment window (parameter information).

   The Detailed Information dialog box is displayed.

5. In the OpenTP1 Environment window (parameter information), click the **Next** button.

   The OpenTP1 Environment Setup window (Output) is displayed.

6. In the OpenTP1 Environment Setup window (Output), select the OpenTP1 service name whose environment you want to set up, and then select the **Execute Batch file** check box.

   Once you select the **Execute Batch file** check box, clicking the **Execution** button creates the definition files and OpenTP1 file system.

7. In the OpenTP1 Environment Setup window (Output), click the **Execution** button.

   A template to which the definition type, model, and parameter values have been applied is stored. A batch file is executed and the definition files and OpenTP1 file system are created.

8. To check the execution results of the batch file, in the OpenTP1 Environment Setup window (Output), click the **View Log** button.

   The Log File dialog box is displayed.

9. In the OpenTP1 Environment Setup window (Output), click the **Close** button.

   The OpenTP1 Environment Setup window (Output) closes and setup of the OpenTP1 environment is completed.

The following subsections provide the details of the windows and dialog boxes used for environment setup.

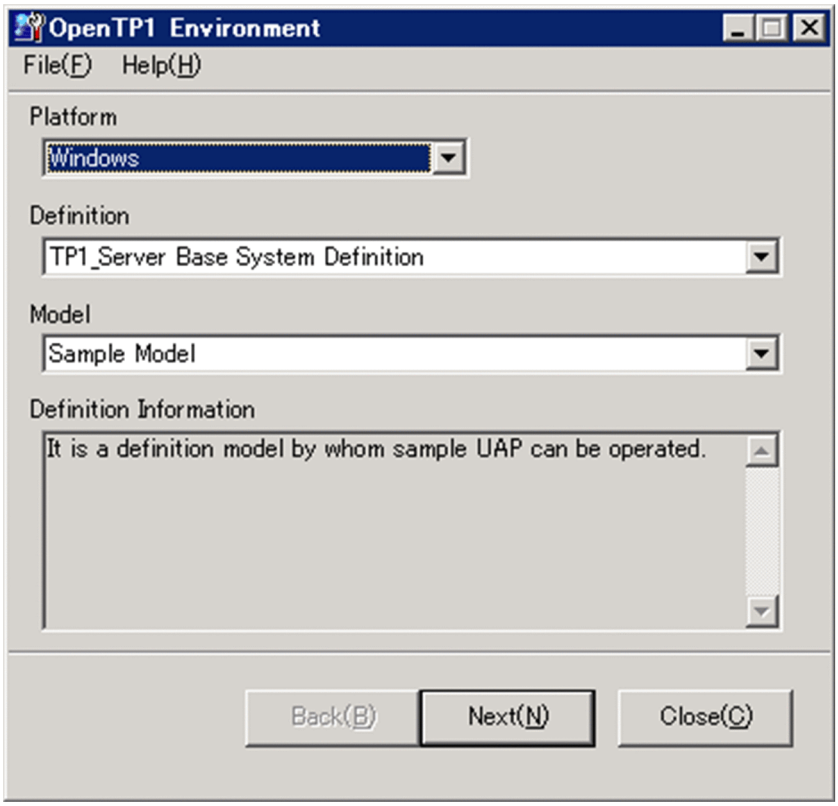
(1) OpenTP1 Environment window

You specify a definition type and model in the OpenTP1 Environment window.

To display the OpenTP1 Environment window, from the **Start** menu, choose **Programs**, **OpenTP1**, and then **OpenTP1 Environment**.

The following shows the OpenTP1 Environment window.

Figure 6-2: OpenTP1 Environment window



Menu bar

File menu

Choosing **Close** from the **File** menu closes the OpenTP1 Environment window.

Help menu

Choosing **About** from the **Help** menu displays the Version dialog box. This dialog box displays information about the OpenTP1 administrator that was entered when OpenTP1 was installed.

Platform drop-down list

Windows is displayed. There is no need to specify this item.

Definition drop-down list

You select one of the following types of definitions:

- **TP1_Server Base System Definition**

  Select this definition type to start OpenTP1. This definition type enables you to specify parameters required for starting OpenTP1 (specific operands in some of the definitions, such as system common definition RAP-processing listener service definition).

  If you wish to specify a user service definition, select **User Service Definition**.

- **User Service Definition**

  Select this definition type to start a user server. This definition type enables you to specify parameters required for starting a user server (some of the operands in the user service definition). Note that you must select **TP1_Server Base System Definition** to create definition files before you select **User Service Definition**.

Model drop-down list

You can select a model definition template. Available models depend on the definition type selected from the **Definition** drop-down list. The following table lists the available models.

Table 6-7: List of available models

Selected definition	Available models	Description
TP1_Server Base System Definition	**Sample Model**	Enables you to start the OpenTP1 service and use the sample UAP.
	Sample Model(RAP)	Enables you to start the OpenTP1 service and use the RAP-processing server.
	Large Model	Model for a large-scale system environment.
User Service Definition	**SPP Sample Model**	Model for the sample SPP.
	SPP.NET Sample Model	Model for the sample SPP.NET.
	SPP.NET Sample Model(.NET Interface)	Model for the sample SPP.NET using the .NET interface.
	SUP Sample Model	Model for the sample SUP.
	SUP.NET Sample Model	Model for the sample SUP.NET.

Selected definition	Available models	Description
	SUP.NET Sample Model(.NET Interface)	Model for the sample SUP.NET using the .NET interface.

Definition Information

Displays a description of the model selected from the **Model** drop-down list.

Next button

Displays the OpenTP1 Environment window (parameter information). For details about the OpenTP1 Environment window (parameter information), see *(2) OpenTP1 Environment window (parameter information)*.

Close button

Terminates the GUI environment setup without saving the specified settings.

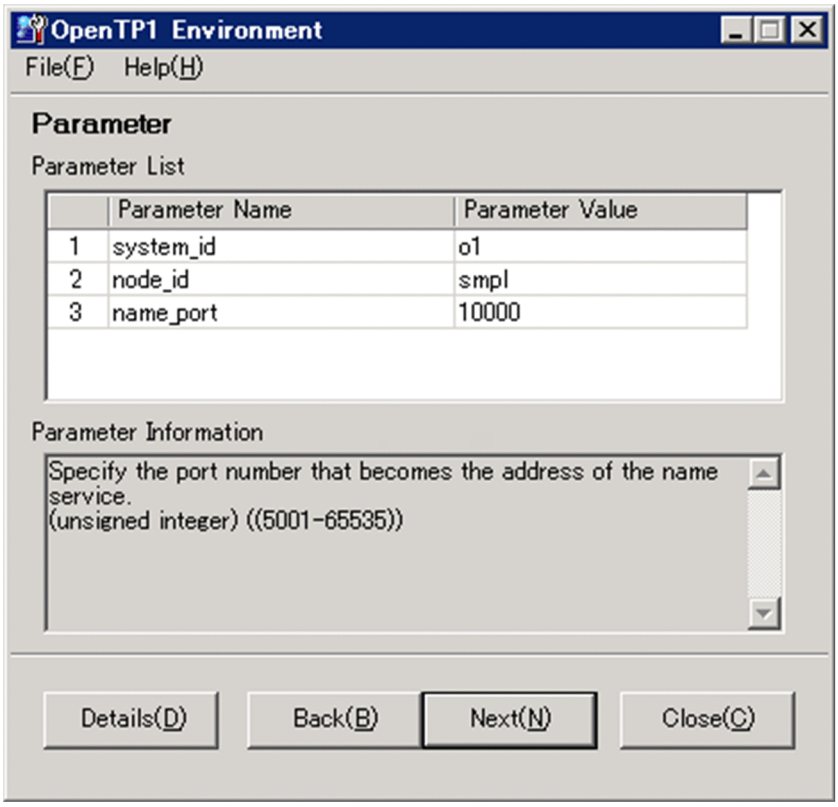
(2) OpenTP1 Environment window (parameter information)

You use the OpenTP1 Environment window (parameter information) to enter parameter values.

To display the OpenTP1 Environment window (parameter information), click the **Next** button at the bottom of the OpenTP1 Environment window.

The following figure shows the OpenTP1 Environment window (parameter information).

Figure 6-3: OpenTP1 Environment window (parameter information)



Parameter List

Displays the names of and values for parameters appropriate to the definition type and model selected in the OpenTP1 Environment window. You can change parameter values as appropriate to the OpenTP1 environment. Note that only entries in **Parameter Value** can be changed in **Parameter List**.

For details about the parameters, see Chapter *5. System Definitions* and the descriptions of the operands with the same names in the manual *OpenTP1 Version 7 System Definition*.

Parameter Information

Selecting the **Parameter Value** cell for a parameter displays a description of that parameter.

Details button

Displays the Detailed Information dialog box. For details about the Detailed Information dialog box, see *(3) Detailed Information dialog box*.

Back button

Returns to the OpenTP1 Environment window.

Next button

Displays the OpenTP1 Environment Setup window (Output). For details about the OpenTP1 Environment Setup window (Output), see *(4) OpenTP1 Environment Setup window (Output)*.

Close button

Terminates GUI environment setup without saving the specified settings.

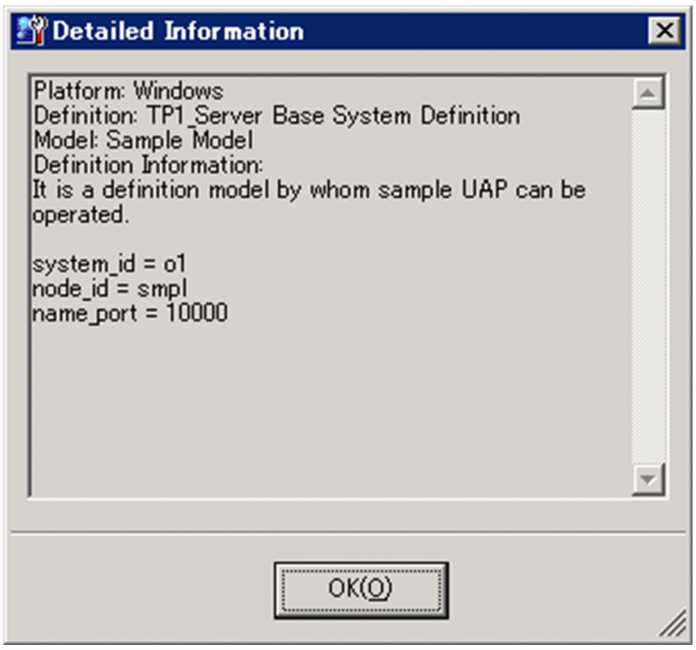
(3) Detailed Information dialog box

You use the Detailed Information dialog box to check the settings specified in the OpenTP1 Environment window and OpenTP1 Environment window (parameter information).

To display the Detailed Information dialog box, click the **Details** button in the OpenTP1 Environment window (parameter information).

The following figure shows the Detailed Information dialog box.

Figure 6-4: Detailed Information dialog box



Displayed information

Displays the settings specified in the OpenTP1 Environment window and OpenTP1 Environment window (parameter information).

OK button

Closes the Detailed Information dialog box.

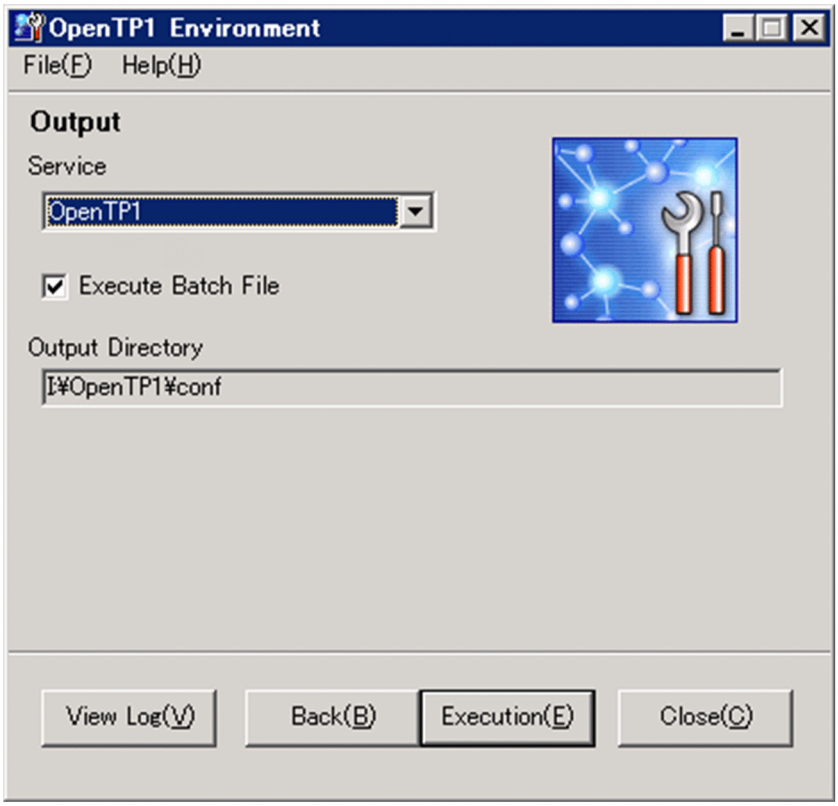
(4) OpenTP1 Environment Setup window (Output)

In the OpenTP1 Environment Setup Window (Output), you select an OpenTP1 service whose environment you want to set up, and specify a folder in which to store the template.

To display the OpenTP1 Environment Setup window (Output), click the **Next** button in the OpenTP1 Environment window (parameter information).

The following figure shows the OpenTP1 Environment Setup window (Output).

Figure 6-5: OpenTP1 Environment Setup window (Output)



Service drop-down list

Select the OpenTP1 service for which a template is to be stored. The drop-down

list displays the following OpenTP1 services that have been set up in Windows:

- **OpenTP1**

  This service is created automatically after OpenTP1 has been installed. When you select **OpenTP1**, the template is stored in `%DCDIR%\conf`.

- **OpenTP1_***XXXX*

  This is a multiOpenTP1 service, where *XXXX* indicates the multiOpenTP1 identifier. When you select **OpenTP1_***XXXX*, the template is stored in the multiOpenTP1 folder.

- **Others**

  Enables you to select a desired storage location for the template.

Execute Batch file check box

When you select this check box, the batch file is executed automatically when the template is stored in order to create the definition files and OpenTP1 file system.

Output Directory

Displays or enters the storage folder, as determined by the selection made in the **Service** drop-down list.

- When **OpenTP1** or **OpenTP1_***XXXX* is selected

  The selected OpenTP1's `%DCDIR%\conf` is displayed.

- When **Others** is selected

  The user enters the path of the folder in which the template is to be stored.

Notes about specifying the storage folder

Not only the parameters listed in **Parameter List** but also the `DCDIR` environment variable's value are defined in the template provided by OpenTP1. `@DCDCDIR@` is set temporarily for the value of the `DCDIR` environment variable, because the user can enter a desired storage destination by selecting **Others** from the **Service** drop-down list. However, environment setup cannot be performed while this environment variable is set to `@DCDCDIR@`, so you must change this value to the correct `%DCDIR%` value for the actual environment being used.

View Log button

Displays the Log File dialog box. For details about the Log File dialog box, see *(5) Log File dialog box*.

Back button

Returns to the OpenTP1 Environment window (parameter information).

Execution button

Stores the template at the folder path displayed in **Output Directory**. If the **Execute Batch file** check box is selected, the batch file is executed.

Once the batch file executes, a dialog box is displayed when the processing is completed. If the processing fails, an error message is displayed in the dialog box. For details about the error messages, see the manuals *OpenTP1 Version 7 Messages*.

Close button

Terminates GUI environment setup.

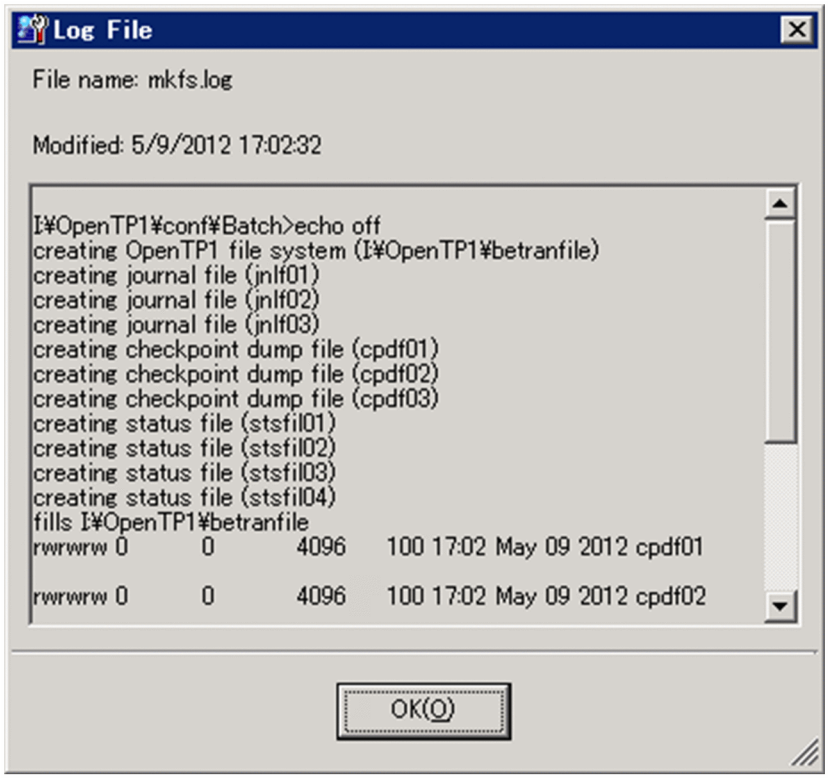
(5) Log File dialog box

You use the Log File dialog box to check the batch file execution results.

To display the Log File dialog box, click the **View Log** button in the OpenTP1 Environment Setup window (Output).

The following figure shows the Log File dialog box.

Figure 6-6: Log File dialog box



File name

Displays the name of the log file that contains the batch file execution results.

Modified

Displays the date and time the batch file executed.

Displayed information

Displays the batch file execution results.

OK button

Closes the Log File dialog box.

6.7.2 GUI for starting and terminating OpenTP1

This section explains the use of the GUI to start and terminate OpenTP1. You can also use the GUI to specify a startup option (startup mode) or termination option (termination mode) before starting or terminating OpenTP1.

If the required definition files and OpenTP1 file system have been created, you can use the GUI to start and terminate OpenTP1 without using the GUI to set up the environment.

To use the GUI to start OpenTP1:

1. From the **Start** menu, choose **Programs**, **OpenTP1**, and then **OpenTP1 Control**.

   The OpenTP1 Control window is displayed.

2. In the OpenTP1 Control window, select the OpenTP1 service that you wish to start.

3. To specify an OpenTP1 startup option (startup mode), click the **Option** button in the OpenTP1 Control window.

   The Option dialog box is displayed.

4. In the OpenTP1 Control window, click the **Start** button.

   The selected OpenTP1 service starts.

5. In the OpenTP1 Control window, click the **Refresh** button.

   The selected OpenTP1 service is updated to its most recent status.

6. In the OpenTP1 Control window, click the **Close** button.

   The OpenTP1 Control window closes and OpenTP1 start processing is completed.

To use the GUI to terminate OpenTP1:

1. From the **Start** menu, choose **Programs**, **OpenTP1**, and then **OpenTP1 Control**.

   The OpenTP1 Control window is displayed.

2. In the OpenTP1 Control window, select the OpenTP1 service that you wish to stop.

3. To specify an OpenTP1 termination option (termination mode), click the **Option** button in the OpenTP1 Control window.

   The Option dialog box is displayed.

4. In the OpenTP1 Control window, click the **Stop** button.

   The selected OpenTP1 service stops.

5. In the OpenTP1 Control window, click the **Refresh** button.

   The selected OpenTP1 service is updated to its most recent status.

6. In the OpenTP1 Control window, click the **Close** button.

The OpenTP1 Control window closes and OpenTP1 stop processing is completed.

Note:

If OpenTP1 issues the `KFCA00715-E` message, selecting the **Stop** button might not stop the OpenTP1 service. In such a case, use the Windows Services dialog box, or the `net stop` command, to stop the OpenTP1 service. For details about the termination methods, see *6.3 Terminating OpenTP1*.

The following subsections provide the details of the windows and dialog boxes used to start and terminate OpenTP1.

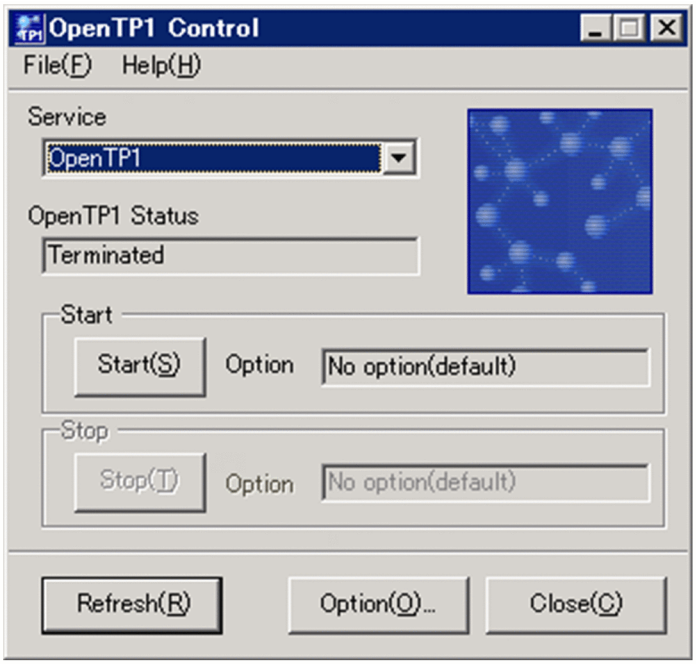
(1) OpenTP1 Control window

You use the OpenTP1 Control window to start or stop a selected service.

To display the OpenTP1 Control window, from the **Start** menu, choose **Programs**, **OpenTP1**, and then **OpenTP1 Control**.

The following figure shows the OpenTP1 Control window.

Figure 6-7: OpenTP1 Control window



For details about the menu bar, see *6.7.1(1) OpenTP1 Environment window*.

Service drop-down list

Displays the names of the OpenTP1 services. The displayed names are the same as in the **Service** drop-down list in the OpenTP1 Environment Setup window (Output). For details about the **Service** drop-down list in the OpenTP1 Environment Setup window (Output), see *6.7.1(4) OpenTP1 Environment Setup window (Output)*.

OpenTP1 Status

Displays the status of the OpenTP1 service selected from the **Service** drop-down list.

You can obtain the details of the displayed OpenTP1 start or termination status by checking the message log file and Event Viewer.

Start button

Starts the OpenTP1 service selected from the **Service** drop-down list. **Option** displays the OpenTP1 startup option (startup mode). For details about the OpenTP1 startup options (startup mode), see *(2) Option dialog box*.

Stop button

Stops the OpenTP1 service selected from the **Service** drop-down list. **Option** displays the OpenTP1 termination option (termination mode). For details about the OpenTP1 termination options (termination modes), see *(2) Option dialog box*.

Refresh button

Updates the OpenTP1 service displayed in the **Service** drop-down list to its most recent status.

Option button

Displays the Option dialog box. For details about the Option dialog box, see *(2) Option dialog box*.

Close button

Terminates GUI specification of OpenTP1 start and stop settings.

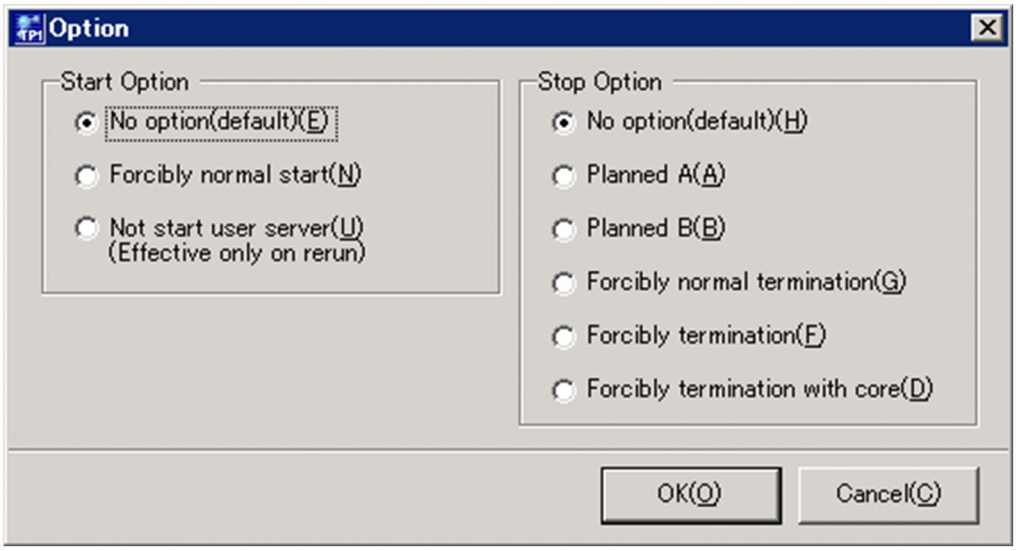
(2) Option dialog box

You use the Option dialog box to specify the OpenTP1 startup option (startup mode) or termination option (termination mode). For details about the modes, see the manual *OpenTP1 Version 7 Operation*.

To display the Option dialog box, click the **Option** button in the OpenTP1 Control window.

The following figure shows the Option dialog box.

Figure 6-8: Option dialog box



Start Option

Select an OpenTP1 startup option (startup mode) from the following radio buttons:

- **No option(default)**

  Starts OpenTP1 in the normal startup mode if the previous online session terminated normally and there is no information to be inherited or an OpenTP1 session is to be started. If OpenTP1 cannot be started in the normal startup mode, it is restarted in order to inherit the previous processing.

- **Forcibly normal start**

  Starts OpenTP1 forcibly in the normal startup mode regardless of the previous termination option (termination mode).

- **Not start user server**

  Does not start the user server when OpenTP1 is restarted.

Stop Option

Select an OpenTP1 termination option (termination mode) from the following radio buttons:

- **No option(default)**

  Terminates OpenTP1 normally.

- **Planned A**

Stops accepting any new service requests, completes processing of all service requests in the schedule queue, and then terminates OpenTP1.

- **Planned B**

  Waits for completion of the service that is currently being processed and then terminates OpenTP1.

- **Forcibly normal termination**

  Stops accepting any new service requests in the same manner as in the normal termination mode even if a server terminated abnormally during operation, completes processing of all service requests in the schedule queue, and then terminates OpenTP1.

  This mode forcibly terminates OpenTP1 normally regardless of the user server status.

- **Forcibly termination**

  Forcibly terminates OpenTP1 immediately without waiting for completion of the current service that is being executed.

- **Forcibly termination with core**

  Collects core files and shared memory information for all active processes and then forcibly terminates OpenTP1.

OK button

Applies the selected option and then closes the dialog box.

Cancel button

Closes the dialog box without applying the selected option.

Chapter

7. Operation Commands

The available operation commands differ between OpenTP1 for Windows and OpenTP1 for UNIX. This chapter provides a list of the operation commands for purposes of describing the differences in the commands between OpenTP1 for Windows and OpenTP1 for UNIX. This chapter also provides detailed explanations of selected operation commands.

For information common to OpenTP1 for Windows and OpenTP1 for UNIX, see the OpenTP1 series manuals.

List of operation commands

The available commands differ between OpenTP1 for Windows and OpenTP1 for UNIX. Limitations, such as unsupported options and settings, apply to some commands. If you are a new user of OpenTP1 for Windows, we recommend that you use the list provided in this section to check for differences from OpenTP1 for UNIX.

Once you have checked this list, see the manual *OpenTP1 Version 7 Operation* for those commands common to OpenTP1 for Windows and OpenTP1 for UNIX. For those commands that differ between them, see the applicable subsections in *Details of commands common to OpenTP1 for Windows and OpenTP1 for UNIX* in this chapter, as necessary. For details about the commands specific to OpenTP1 for Windows, see the applicable subsections in *Details of the commands specific to OpenTP1 for Windows* in this chapter.

Note:

The provided list does not include the following commands:

- Message queuing (TP1/Message Queue) commands
- TP1/Extension for .NET Framework commands

The legend for the list that follows is shown below.

Legend:

S: Same (command is supported identically in OpenTP1 for Windows and OpenTP1 for UNIX)

W: Windows-specific (command is specific to OpenTP1 for Windows)

D: Differences (command has differences in OpenTP1 for Windows from OpenTP1 for UNIX or some limitations apply to OpenTP1 for Windows)

N: Not supported for OpenTP1 for Windows

--: Not applicable

The following table lists and describes the differences in the operation commands between OpenTP1 for Windows and OpenTP1 for UNIX.

Table 7-1: Differences in operation commands between OpenTP1 for Windows and OpenTP1 for UNIX

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
System management	Register OpenTP1 into the OS or delete OpenTP1 from the OS	`dcsetup`	N	Not supported. OpenTP1 for Windows is registered into the OS by the installer. Therefore, if you execute this command after OpenTP1 has already been installed, the OpenTP1 environment might be corrupted. If you execute this command by mistake, reinstall OpenTP1.
	Restart the process service and update it from the definitions	`dcreset`	N	Not supported. For details about changing the system definition, see *4.3.2 Changing definitions*.
	Reserve and release resources for OpenTP1 internal control	`dcmakeup`	N	Not supported.
	Start OpenTP1	`dcstart`	N	Not supported. For details about how to start OpenTP1 for Windows, see *6.2 Starting OpenTP1*.
	Terminate OpenTP1	`dcstop`	S	For details about how to terminate OpenTP1 for Windows, see *6.3 Terminating OpenTP1*.
	Start and terminate acquisition of system statistical information	`dcstats`	D	`ist` and `xat` are not supported as `-k` option values.
	Start a multi-node area or subarea	`dcmstart`	N	Not supported.
	Terminate a multi-node area or subarea	`dcmstop`	N	
	Execute an OpenTP1 command from a scenario template	`dcjcmdex`	S	--
	Specify a system definition operand	`dcjchconf`	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Update a domain definition file	dcjnamch	S	--
	Display the status of one or more OpenTP1 nodes	dcndls	N	Not supported.
	Display the status of shared memory	dcshmls	S	--
	Display the status of temporary close processing	rpcstat	S	--
	Redirect standard output and standard error	prctee	N	Not supported. Specify the output format in the redirect_file operand in the system environment definition. For details about the redirect_file operand, see *System environment definition* in *5.3 Details of system definitions*.
	Terminate and restart the prctee process	prctctrl	N	Not supported.
	Acquire maintenance data	dcrasget	D	The -c option is not supported. The maintenance data storage location differs between OpenTP1 for Windows and OpenTP1 for UNIX. For details, see *dcrasget (acquires maintenance data)*.
	Edit and output system statistical information to standard output in real time	dcreport	S	--
	Delete troubleshooting information	dccspool	S	--
	Check system definitions	dcdefchk	D	This command does not check access permissions at the NTFS level during local checking. Therefore, if a folder for which there is no access permission is specified in the system definitions, operations such as file creation will fail.

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Display product information	dcpplist	D	There are differences in the output format of the names of products that are running in the environment configured in the OpenTP1 directory. For details, see *dcpplist (displays product information)*.
Server management	Start one or more servers	dcsvstart	S	--
	Terminate one or more servers	dcsvstop	S	--
	Display the status of one or more servers	prcls	D	Specify for the module name in the -l option the module operand value specified in the user service definition.
	Display the search path names for the user server and for the commands started from the user server	prcpathls	S	--
	Change one or more search path names for the user server and for the commands started from the user server	prcpath	D	Even when a user server path has been changed, %DCDIR%\bin is always added at the beginning.
	Forcibly terminate one or more OpenTP1 processes	prckill	S	There are no differences in this command between OpenTP1 for Windows and OpenTP1 for UNIX. However, only a value of 3, 6, or 15 is permitted as the signal number for forced termination in the prc_abort_signal operand in the user service definition. For details about the prc_abort_signal operand, see *User service definition* in *5.3 Details of system definitions*.
Schedule management	Display scheduling status	scdls	D	The -b option is not supported.
	Shut down scheduling	scdhold	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Restart scheduling	scdrles	S	--
	Change the number of processes	scdchprc	S	--
	Terminate and restart processes	scdrsprc	S	--
Transaction management	Display transaction status	trnls	D	OpenTP1_ISM cannot be specified as the resource manager in the -R option.
	Commit transactions	trncmt	S	--
	Roll back transactions	trnrbk	S	--
	Forcibly terminate transactions	trnfgt	S	--
	Start and terminate acquisition of transaction statistical information	trnstics	S	--
	Delete undetermined-transaction information files	trndlinf	S	--
	Display undetermined-transaction information for OSI TP communication	tptrnls	N	Not supported.
XA resource management	Display XAR event trace information	xarevtr	S	--
	Display the status of a XAR file	xarfills	S	--
	Change the status of a XAR transaction	xarforce	S	--
	Shut down the XA resource service	xarhold	S	--
	Create a XAR file	xarinit	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Display XAR transaction information	xarls	S	--
	Release the XA resource service from shutdown status	xarrles	S	--
	Delete a XAR file	xarrm	S	--
Lock management	Display lock information	lckls	S	--
	Display pool information from the lock control table	lckpool	S	--
	Delete deadlock and timeout information files	lckrminf	S	--
Name management	Check OpenTP1 startup and clear cache	namalivechk	S	--
	Register and delete the domain representative schedule service	namdomainsetup	N	Not supported.
	Change the domain configuration (using the system common definition)	namndchg	S	--
	Change the domain configuration (using domain definition files)	namchgfl	S	--
	Forcibly disable start notification information	namunavl	S	--
	Display OpenTP1 server information	namsvinf	D	If the address cannot be resolved for a node name specified in the all_node or all_node_ex operand in the system common definition, command execution might take a long time.

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Manipulate the RPC suppression list	namblad	S	--
Message log management	Display the contents of a message log file	logcat	S	--
	Change the setting for the real-time output facility for message logs	logcon	S	--
Audit log management	Set up the enviroment for the audit log facility	dcauditsetup	S	--
OpenTP1 file management	Initialize an OpenTP1 file system	filmkfs	S	--
	Display the status of an OpenTP1 file system	filstatfs	D	The -w option is not supported.
	Display the contents of an OpenTP1 file system	fills	D	The process ID of a locked process cannot be displayed even if the command is executed with the -H and -L options both specified. For details, see *fills (displays the contents of the OpenTP1 file system)*.
	Back up an OpenTP1 file system	filbkup	S	--
	Restore an OpenTP1 file system	filrstr	S	--
	Change an OpenTP1 file group	filchgrp	N	Not supported.
	Change the access permission mode for an OpenTP1 file	filchmod	N	
	Change an OpenTP1 file owner	filchown	N	
Status file management	Create and initialize a status file	stsinit	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Display the status of status files	stsls	S	--
	Display the contents of a status file	stsfills	S	--
	Open a status file	stsopen	S	--
	Close a status file	stsclose	S	--
	Delete a status file	stsrm	S	--
	Swap status files	stsswap	S	--
Journal file management	Initialize a journal file	jnlinit	S	--
	Display journal file information	jnlls	D	The following settings are not supported: • -j option's jar value • -r option
	Display journal file information that was read during restart	jnlrinf	S	--
	Open journal files	jnlopnfg	D	The following settings are not supported: • -j option's jar value • -r option • -e option
	Close journal files	jnlclsfg	D	
	Allocate a journal physical file	jnladdpf	S	--
	Delete a journal physical file	jnldelpf	S	--
	Swap journal files	jnlswpfg	D	The following settings are not supported: • -j option's jar value • -r option
	Delete a journal file	jnlrm	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Change the status of journal files	jnlchgfg	D	The following settings are not supported: • -j option's jar value • -S option
	Unload journal files	jnlunlfg	D	The -j option's jar value is not supported. The -o option (specification of the output destination file name) cannot be omitted. For details, see *jnlunlfg (unloads journal files)*.
	Control the automatic unloading facility	jnlatunl	S	--
	Recover journal files	jnlmkrf	S	--
	Collect journals required for file recovery	jnlcolc	D	The -z option (specification of the output file name) specific to OpenTP1 for Windows cannot be omitted. Specification of a file name cannot be omitted in the command argument. The -k option's i value is not supported. For details, see *jnlcolc (collects journals required for file recovery)*.
	Copy unload journal files	jnlcopy	D	The -z option (specification of the output file name) specific to OpenTP1 for Windows cannot be omitted. Specification of an unload journal file name cannot be omitted in the command argument. The -o option is not supported. For details, see *jnlcopy (copies unload journal files)*.
	Display archive status	jnlarls	N	Not supported.

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Edit and output unload journal files or global archive unload journal files	jnledit	D	Specification of a file name cannot be omitted in the command argument. The -o option is not supported. For details, see *jnledit (edits and outputs unload journal files or global archive unload journal files)*.
	Output records from unload journal files or global archive unload journal files	jnlrput	D	The -z option (specification of the output file name) specific to OpenTP1 for Windows cannot be omitted. Also, specification of a file name cannot be omitted in the command argument. The -q_trn and -o options are not supported. For details, see *jnlrput (outputs records from unload journal files or global archive unload journal files)*.
	Sort and merge unload journal files and global archive unload journal files in time order	jnlsort	D	The -z option (specification of the output file name) specific to OpenTP1 for Windows cannot be omitted. Specification of a file name cannot be omitted in the command argument. This command can be executed only when the -g option is specified. For details, see *jnlsort (sorts and merges unload journal files and global archive unload journal files in time order)*.
	Output statistical information	jnlstts	D	Specification of a file name cannot be omitted in the command argument. For details, see *jnlstts (outputs statistical information)*.
	Output MCF statistics	jnlmcst	D	Specification of a file name cannot be omitted in the command argument. The -o option is not supported. For details, see *jnlmcst (outputs MCF statistics*).

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Forcibly release resource group connection	jnlardis	N	Not supported.
DAM file management	Initialize a physical file	damload	S	--
	Display the status of logical files	damls	S	--
	Add a logical file	damadd	S	--
	Detach a logical file	damrm	S	--
	Logically shut down a logical file	damhold	S	--
	Release a logical file from shutdown status	damrles	S	--
	Delete a physical file	damdel	S	--
	Back up a physical file	dambkup	D	The -s option is not supported.
	Restore a physical file	damrstr	D	
	Restore logical files	damfrc	S	--
	Set a threshold for the number of cache blocks	damchdef	S	--
	Acquire the number of cache blocks	damchinf	S	--
TAM file management	Initialize a TAM file	tamcre	S	--
	Display the status of TAM tables	tamls	S	--
	Add a TAM table	tamadd	S	--
	Detach a TAM table	tamrm	S	--
	Logically shut down a TAM table	tamhold	S	--
	Release a TAM table from shutdown status	tamrles	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Load a TAM table	tamload	S	--
	Unload a TAM table	tamunload	S	--
	Delete a TAM file	tamdel	S	--
	Back up a TAM file	tambkup	D	The -s option is not supported.
	Restore a TAM file	tamrstr	D	
	Recover a TAM file	tamfrc	S	--
	Convert TAM locked resource names	tamlckls	S	--
	Display synonym information for a hash-format TAM file and table	tamhsls	S	--
Message queue file management	Display the status of queue groups	quels	S	--
	Allocate a physical file for the message queue	queinit	S	--
	Delete a physical file for the message queue	querm	S	--
Resource manager management	Display resource manager information	trnlsrm	S	--
	Register resource managers	trnlnkrm	D	Neither OpenTP1_ISM nor OpenTP1_XATMI can be specified as a resource manager in an option. There is no need to specify the -P option. For details, see *trnlnkrm (registers resource managers)*.
	Create a transaction control object file	trnmkobj	D	OpenTP1_ISM cannot be specified as a resource manager in the -R option. The -P option is not supported. For details, see *trnmkobj (creates a transaction control object file)*.

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Trace management	Edit and output UAP trace information	`uatdump`	D	UAP trace files are output by default. For details about the output location and names of the UAP trace files, see the descriptions of the files output by OpenTP1 in the manual *OpenTP1 Version 7 Operation*.
	Merge RPC trace information	`rpcmrg`	D	The `-z` option (specification of the output file name) specific to OpenTP1 for Windows cannot be omitted. For details, see *rpcmrg (merges RPC trace information)*.
	Output RPC trace information	`rpcdump`	D	If you specify the `-r` option, you must also specify the output file name. For details, see *rpcdump (outputs RPC trace information)*.
	Output a shared memory dump	`usmdump`	S	--
Remote API management	Set up an execution environment for the remote API facility	`rapsetup`	S	--
	Automatically generate the definitions used for the remote API facility	`rapdfgen`	S	--
	Display the status of a RAP-processing listener and server	`rapls`	S	--
Management of performance verification traces	Edit and output trace information files	`prfed`	S	--
	Get trace information files	`prfget`	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Real-time statistics service management	Edit and output RTS log files	rtsedit	D	**** cannot be specified in the -v option if you want to output only the real-time statistics that have been acquired for each processing other than the service.
	Output real-time statistics to standard output	rtsls	D	
	Set up an execution environment for the real-time statistics service	rtssetup	S	--
	Change settings for real-time statistics	rtsstats	S	--
Connection management	Display connection status	mcftlscn	S	--
	Establish connections	mcftactcn	S	--
	Release connections	mcftdctcn	S	--
	Change connections	mcftchcn	S	--
	Display network status	mcftlsln	S	--
	Start acceptance of server-type connection establishment requests	mcftonln	S	--
	End acceptance of server-type connection establishment requests	mcftofln	S	--
	Display concurrent message processing status	mcftlstrd	S	--
Application management	Display application status	mcfalsap	S	--
	Shut down applications	mcfadctap	S	--
	Release applications from shutdown status	mcfaactap	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Initialize the application abnormal termination count	mcfaclcap	S	--
	Display application timer start requests	mcfalstap	S	--
	Delete application timer start requests	mcfadltap	S	--
Application operation support	Start an application program	mcfuevt	S	--
Logical terminal management	Display logical terminal status	mcftlsle	S	--
	Shut down logical terminals	mcftdctle	S	--
	Release logical terminals from shutdown status	mcftactle	S	--
	Skip the first message in the message queue for a logical terminal	mcftspqle	S	--
	Place output queue processing for a logical terminal in hold status	mcfthldoq	S	--
	Release output queue processing for a logical terminal from hold status	mcftrlsoq	S	--
	Delete output queues for logical terminals	mcftdlqle	S	--
	Start acquisition of logical terminal-related message journals	mcftactmj	S	--
	End acquisition of logical terminal-related message journals	mcftdctmj	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
	Forcibly end continuous inquiry-response processing for a logical terminal	mcftendct	S	--
	Start alternate sending	mcftstalt	S	--
	End alternate sending	mcftedalt	S	--
Service group management	Display the status of service groups	mcftlssg	S	--
	Shut down service groups	mcftdctsg	S	--
	Release service groups from shutdown status	mcftactsg	S	--
	Place input queue processing for service groups in hold status	mcfthldiq	S	--
	Release input queue processing for service groups from hold status	mcftrlsiq	S	--
	Delete input queues for service groups	mcftdlqsg	S	--
Service management	Display service status	mcftlssv	S	--
	Shut down services	mcftdctsv	S	--
	Release services from shutdown status	mcftactsv	S	--
Session management	Start a session	mcftactss	S	--
	End a session	mcftdctss	S	--
Buffer management	Display the usage status of a buffer group	mcftlsbuf	S	--
Map management	Change the path name of a map file	dcmapchg	S	--
	Display the loaded resources of a map file	dcmapls	S	--

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Queue management	Copy the contents of the input/output queues	mcftdmpqu	S	--
MCF trace acquisition management	Forcibly swap MCF trace files	mcftswptr	S	--
	Start acquisition of MCF trace information	mcftstrtr	S	--
	End acquisition of MCF trace information	mcftstptr	S	--
Management of MCF statistics	Edit MCF statistics	mcfreport	N	Not supported.
	Output MCF statistics	mcfstats	N	
MCF communication service management	Partially stop an MCF communication service	mcftstop	S	--
	Partially start an MCF communication service	mcftstart	S	--
	Reference the status of an MCF communication service	mcftlscom	S	--
User timer management	Display the status of user timer monitoring	mcftlsutm	S	--
MultiOpenTP1 management	Check the setup status of multiOpenTP1s	dcdls	W	This command is specific to OpenTP1 for Windows. For details, see *dcdls (checks the setup status of multiOpenTP1s)*.
	Create a multiOpenTP1 console	dcmakecon	W	This command is specific to OpenTP1 for Windows. For details, see *dcmakecon (creates a multiOpenTP1 console)*.
	Set up a multiOpenTP1 or cancel setup of a multiOpenTP1	dcsetupml	W	This command is specific to OpenTP1 for Windows. For details, see *dcsetupml (sets up a multiOpenTP1 or cancels setup of a multiOpenTP1)*.

Classification/purpose		Operation command	Support by OpenTP1 for Windows	Difference from OpenTP1 for UNIX
Resource manager management	Create a transaction control DLL	`njsmkdll`	W	This command is specific to OpenTP1 for Windows. For details, see *njsmkdll (creates a transaction control DLL)*.
System management	Start OpenTP1	`ntbstart`	W	This command is specific to OpenTP1 for Windows. For details, see *ntbstart (starts OpenTP1)*.
	Output the contents of a specified file to the console	`ntbtail`	W	This command is specific to OpenTP1 for Windows. For details, see *ntbtail (outputs the contents of a specified file to the console)*.
	Output the standard output redirect file to the console	`tp1console`	W	This command is specific to OpenTP1 for Windows. For details, see *tp1console (outputs the standard output redirect file to the console)*.

Format of command explanations

This section explains the format used to describe the commands.

In this manual, only the parts that differ between OpenTP1 for Windows and OpenTP1 for UNIX are explained. For the parts that are common to both, see the manual *OpenTP1 Version 7 Operation*.

Format

Explains the command format. Options and arguments that are specific to OpenTP1 for Windows and that differ from those for OpenTP1 for UNIX are indicated in **bold**.

Function

Explains the function of the command.

Options

Explains the options of the command.

Command arguments

Explains the arguments of the command.

Output format

Explains the output format of the command.

Output messages

Explains the messages that are issued by the command.

Notes

Provides notes about the command.

Details of commands common to OpenTP1 for Windows and OpenTP1 for UNIX

The following subsections provide the details of the differences in the commands common to OpenTP1 for Windows and OpenTP1 for UNIX.

Note:

If you start a command prompt to execute OpenTP1 commands on a system running Windows Vista or later, or Windows Server 2008 or later, choose **Run as administrator**. If you execute an OpenTP1 command from a regular command prompt, the OpenTP1 command might not function correctly.

dcpplist (displays product information)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Output format

```
#    Product ID          Version    Product Name
---  ------------------- --------- ------------------------
aaa  bb...bb              cc...cc       dee...ee
```

- *d*: If the product is installed in the same folder as TP1/Server Base, an asterisk (*) is displayed; if not, a single-byte space (∆) is displayed.
- *ee...ee*: Product name (up to 50 bytes)

There are no other differences.

Notes

Some information about OpenTP1-related products and Hitachi open middleware products is also displayed.

dcrasget (acquires maintenance data)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

`dcrasget [-g] [-l]` ***acquired-data-directory***

Command arguments

- *acquired-data-directory* ~<path name>

  Specifies the folder under which the following folder is to be created (which is where the maintenance data will be stored):

  `dcrasget.HOSTNAME.`*mmddHHMMSS*`.`*AAA*

  `HOSTNAME`: Default host name

  mmddHHMMSS: month, day, hour, minute, second

  AAA: Data acquisition unit:

  `SAV`: Information under the `%DCDIR%\spool\save` folder

  `SPL`: Information under the `%DCDIR%\spool` folder (excluding `save`)

  `TMP`: Information under the `%DCDIR%\tmp` folder

  `CNF`: Information about the relationships among definitions, such as the `%DCDIR%\conf` folder

  `INF`: Information such as information files

  There are no other differences.

fills (displays the contents of the OpenTP1 file system)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Output format

■ When the `-H` and `-L` options are specified

```
MODE     UID     GID    PID    L  TIME    FILE
aabbcc   dd...dd  ee...ee  pppp   q  hh...hh  ii...ii
```

- *pppp*: Process ID of the locking process

  In OpenTP1 for Windows, `0` is always displayed.

There are no other differences.

jnlcolc (collects journals required for file recovery)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlcolc -k recovery-type -z output-file-name [-f][-l][-n][-m]
        [-i online-backup-information-name][-c key]
         file-name[[△file-name]...]
```

Function

This command extracts from specified files the journal records needed to recover a DAM or TAM file and then outputs them to a file.

Options

- `-k` *recovery-type*

  `i` (ISAM file) cannot be specified as the type of file to be recovered. There are no other differences.

- `-z` *output-file-name* ~<path name>

  Specifies the name of the file to which the execution results are to be output. This option cannot be omitted.

Command arguments

- *file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

jnlcopy (copies unload journal files)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlcopy -z output-file-name [-t[begin][,end]]
        [-j record-type[record-type]...]
         unload-journal-file-name
        [[∆unload-journal-file-name]...]
```

Function

This command outputs to a file the contents of a specified unload journal file.

Options

- `-z` *output-file-name* ~<path name>

  Specifies the name of the file to which the execution results are to be output. This option cannot be omitted.

Command arguments

- *unload-journal-file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

jnledit (edits and outputs unload journal files or global archive unload journal files)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnledit [-e editing-type][-t[begin][,end]]
        [-j record-type[record-type]...][-s server-name]
        [-v service-name]
        [-u transaction-global-identifier
                                    [,transaction-branch-identifier]]
        [-m logical-terminal-name[,logical-terminal-name]...]
        [-w digits][-l row][-c]
         file-name[[∆file-name]...]
```

Function

This command edits the contents of a specified unload journal file or global archive unload journal file and then outputs the results to a file.

Command arguments

- *file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

jnlmcst (outputs MCF statistics)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlmcst [-e editing-type][-l row][-i interval]
        [-t[begin][,end]][-m logical-terminal-name]
        [-a application-name]
         file-name
```

Function

This command collects MCF statistics from a specified unload journal file or global archive journal file, edits the collected information, and then outputs the results to a file.

The command outputs the following information:

- MCF statistics related to message reception and transmission

Command arguments

- *file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

jnlrput (outputs records from unload journal files or global archive unload journal files)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlrput -z output-file-name [-t[begin][,end]]
        [-e][-f][-l][-c key]
        [-u transaction-global-identifier
                                    [,transaction-branch-identifier]]
        [-j record-type[record-type]...][-q _rpc]
        [-d any-extraction-condition][-x]file-name[[Δfile-name]...]
```

Function

This command outputs to a file user journal record information, a transaction branch's CPU usage time information, and response statistical information from specified unload journal files or global archive unload journal files.

Options

- `-z` *output-file-name* ~<path name>

  Specifies the name of the file to which the execution results are to be output. This option cannot be omitted.

Command arguments

- *file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

jnlsort (sorts and merges unload journal files and global archive unload journal files in time order)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlsort -z output-file-name [-n node-identifier]-g
        [-i [beginning-journal-server-run-ID]
                                        [,ending-journal-server-run-ID]]
         file-name[△ file-name]...
```

Function

This command merges the contents of specified unload journal files and global archive unload journal files. It outputs the results to a file in the global archive unload journal file format.

Make sure that you execute this command with the `-g` option specified. If the `-g` option is omitted, the command's operation cannot be guaranteed.

Options

- `-z` *output-file-name* ~<path name>

  Specifies the name of the file to which the execution results are to be output. This option cannot be omitted.

- `-g`

  When you execute this command, you must specify this option. There are no other differences.

Command arguments

- *file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

jnlstts (outputs statistical information)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlstts [-e edit-item][-u editing-unit][-l row][-c]
        [-i interval][-t[begin][,end]]
        [-s {server-name|service-name} ]
         file-name
```

Function

This command collects statistical information from a specified unload journal file or global archive unload journal file, edits the collected information, and then outputs the results to a file.

The command outputs the following information:

- System statistical information (information about each component in the system statistics records)
- Statistical information about transactions (information about each transaction in the system statistics records)
- Response statistical information (information about each response time in the system statistics records)
- Statistical information about communication delay time (information about communication delay in the system statistics records)

Command arguments

- *file-name* ~<path name>

  This command argument cannot be omitted. There are no other differences.

Notes

If the communication time is shorter than the accuracy of the OS's system time, data obtained after statistical information about the communication delay time has been edited might result in a negative value, depending on the accuracy of the OS times at the RPC source and target even when the time is not returned to a point in the past. If data in the collection and editing period contains a negative value, the command outputs asterisks (`*`) as the maximum, minimum, and average values for the target editing period.

jnlunlfg (unloads journal files)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
jnlunlfg -j sys [[-f]|[-t[begin][,end]]]
         [-r resource-group][-R run-ID]
         [-g file-group-name]
          -o output-file-name
         [-d service-definition-name][-n]
```

Function

This command unloads a specified file group to an output file and then sets the file group to unloaded status. There are no other differences.

Options

- `-j sys`

  `jar` (archive journal files) cannot be specified as the journal files to be unloaded. There are no other differences.

- `-o` *output-file-name* ~<path name>

  This option cannot be omitted. There are no other differences.

rpcdump (outputs RPC trace information)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
rpcdump [{-r output-file-name|-m}] [{-c|-f}] [-d message-length]
        [-t [begin][,end]]
        [-s service-group-name[,service-name]...]
        [-b node-identifier[,node-identifier]...]
        [-v server-name[,server-name]...]
        [-p process-ID[,process-ID]...]
        [-x xid[,xid]...]
        [-n [beginning-trace-number][,ending-trace-number]]
        [RPC-trace-file-name]
```

Function

This command edits the trace information in a specified RPC trace file and then outputs the results to a file.

Options

- `-r` *output-file-name* ~<path name>

  Specifies the name of the file to which the execution results are to be output. This option cannot be omitted. There are no other differences.

rpcmrg (merges RPC trace information)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
rpcmrg -z output-file-name
        RPC-trace-file-name[[ΔRPC-trace-file-name]...]
```

Function

This command sorts trace information in specified multiple RPC trace files in time order and then outputs the information to a file in RPC trace file format.

If the specified multiple RPC trace files contain the same trace information, the command deletes duplicated information.

Options

- `-z` *output-file-name* ~ <path name>

  Specifies the name of the file to which the execution results are to be output. This option cannot be omitted.

trnlnkrm (registers resource managers)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
trnlnkrm
{[-A
OpenTP1-provided-RM-name-to-be-added[,OpenTP1-provided-RM-name-to-be-added]...]
  [-D
OpenTP1-provided-RM-name-to-be-deleted[,OpenTP1-provided-RM-name-to-be-deleted]...]
   [-a RM-name-that-is-not-provided-by-OpenTP1
    [,RM-name-that-is-not-provided-by-OpenTP1]...
 -s RM-switch-name[,RM-switch-name]...
 -o "RM-related-object-name[ΔRM-related-object-name]..."
 [,"RM-related-object-name[ΔRM-related-object-name]..."]...]
 [-d RM-name-that-is-not-provided-by-OpenTP1
   [,RM-name-that-is-not-provided-by-OpenTP1]...]
|[-n]}
[-C "compiler-option-name[Δcompiler-option-name]..."]
[-B "linkage-option-name[Δlinkage-option-name]..."][-l][-f][-P]
```

Options

- `-A` *OpenTP1-provided-RM-name-to-be-added* ~<1 to 31 alphanumeric characters>

  Neither OpenTP1_ISM nor OpenTP1_XATMI can be specified as the name of an OpenTP1-provided resource manager to be added. There are no other differences.

- `-D` *OpenTP1-provided-RM-name-to-be-deleted* ~<1 to 31 alphanumeric characters>

  Neither OpenTP1_ISM nor OpenTP1_XATMI can be specified as the name of an OpenTP1-provided resource manager to be deleted. There are no other differences.

- `-o` *RM-related-object-name* ~<alphanumeric>

  If you specify multiple objects, enclose the entire list of objects in double quotation marks (`"`). If you specify paths of RM-related object names, use double backslashes (`\\`) as the delimiter. There are no other differences.

- `-C` *compiler-option-name* ~<string of 1 to 512 characters>

  If you specify compiler option names, enclose them in double quotation marks (`"`). There are no other differences.

- `-B` *linkage-option-name* ~<string of 1 to 512 characters>

  If you specify the names of linkage options that are used during library linkage processing, enclose the linkage option names in double quotation marks (`"`). There are no other differences.

- `-P`

  There is no need to specify this option because it is assumed to be specified by default. There are no other differences.

Notes

- You need a C compiler to execute this command. You must specify the folder containing the command in the `PATH` environment variable beforehand because Visual Studio's `cl`, `link`, and `mt` commands execute during execution of this command. You must also specify the `LIB` and `INCLUDE` environment variables to run the compiler successfully. For details, see the Release Notes.
- A manifest file will have been embedded in the executable file created by this command. For details about the manifest file, see the Windows documentation.
- When you execute this command, make sure that you specify the path of Visual Studio at the beginning of the `PATH`, `LIB`, and `INCLUDE` environment variables. If the path of Visual Studio is omitted, unexpected linkage might result.

trnmkobj (creates a transaction control object file)

This manual explains only the parts that differ from OpenTP1 for UNIX. For the parts that are common to both OpenTP1 for Windows and OpenTP1 for UNIX, see the manual *OpenTP1 Version 7 Operation*.

Format

```
trnmkobj -o transaction-control-object-name
        [-R OpenTP1-provided-RM-name[,OpenTP1-provided-RM-name]...]
        [-r
RM-name-that-is-not-provided-by-OpenTP1[,RM-name-that-is-not-provided-by-OpenTP1]...]
        [-C "compiler-option-name[△compiler-option-name]..."]
        [-l]
```

Options

- `-R` *OpenTP1-provided-RM-name* ~<1 to 31 alphanumeric characters>

  OpenTP1_ISM cannot be specified as the name of an OpenTP1-provided resource manager that is accessed from UAPs. There are no other differences.

- `-C` *compiler-option-name* ~<string of 1 to 512 characters>

  If you specify compiler option names, enclose them in double quotation marks (`"`). There are no other differences.

Details of the commands specific to OpenTP1 for Windows

This subsection provides the details of the commands that are specific to OpenTP1 for Windows.

Note:

If you start a command prompt to execute the OpenTP1 commands in Windows Vista or later or Windows Server 2008 or later, choose **Run as administrator**. If you use a regular command prompt, the OpenTP1 commands might not function correctly.

dcdls (checks the setup status of multiOpenTP1s)

Format

```
dcdls [-d OpenTP1-directory]
```

Function

This command displays the OpenTP1 identifiers of the OpenTP1s that are currently set up and the setup folder. You can execute this command whether OpenTP1 is online or offline.

You can execute this command only from the console for the original OpenTP1.

Options

- `-d` *OpenTP1-directory*

  If the specified folder is the setup folder for a multiOpenTP1, this option displays the identifier of that multiOpenTP1.

  If you specify the installation folder of the original OpenTP1, this option displays `Original`.

Output format

When the `-d` option is omitted

```
Identifier      Directory
------          aa...aa
------          bb...bb
```

- *aa...aa*: Original OpenTP1's installation folder
- *bb...bb*: MultiOpenTP1's setup folder

When the `-d` option is specified

```
Identifier  Directory
Original    aa...aa
bbbb        cc...cc
```

- *aa...aa*: Original OpenTP1's installation folder
- *bbbb*: multiOpenTP1's identifier (up to 4 characters)
- *cc...cc*: MultiOpenTP1's setup folder

Output messages

Message ID	Description	Output destination
KFCA26551-E	Command execution failed.	Standard error

Message ID	Description	Output destination
KFCA26552-E	A specified command argument is invalid.	Standard error
KFCA26554-I	Help message	Standard error
KFCA26556-E	The specified folder is not an OpenTP1 folder.	Standard error

Notes

If a serious failure occurs during execution of this command, back up the following data as soon as possible and then contact maintenance personnel:

- `%TEMP%\dcsetupml.log`
- `%TEMP%\trnlnkrm.log`
- `%DCDIR%\spool\errlog\*.*` (`%DCDIR%` indicates the original OpenTP1's installation folder)

dcmakecon (creates a multiOpenTP1 console)

Format

```
dcmakecon [identifier]
```

Function

This command creates a new console that is to be used to execute commands by the OpenTP1 with the specified OpenTP1 identifier. If the identifier is omitted, the command creates a new console that is to be used to execute commands by the original OpenTP1. You can execute this command whether OpenTP1 is online or offline.

You can execute this command only from the console for the original OpenTP1.

In a multiOpenTP1, you execute commands by the original OpenTP1 from the console that is created by execution of this command. Note that none of the following commands can be executed from the console for the multiOpenTP1:

- dcdls
- dcmakecon
- dcsetupml

The environment variables listed in the table below are set at the new console created by execution of this command. These settings are effective only at the corresponding console.

Table 7-2: Environment variable settings in the console

Environment variable	Setting
DCCONFPATH	The same value as the DCCONFPATH environment variable value defined in *OpenTP1-setup-folder-corresponding-to-OpenTP1-identifier*\conf\env is set.
DCDIR	*OpenTP1-setup-folder-corresponding-to-OpenTP1-identifier* is set.
DCUAPCONFPATH	The same value as the DCUAPCONFPATH environment variable value defined in *OpenTP1-setup-folder-corresponding-to-OpenTP1-identifier*\conf\env is set.
INCLUDE	*OpenTP1-setup-folder-corresponding-to-OpenTP1-identifier*\include is added at the beginning.
LIB	*OpenTP1-setup-folder-corresponding-to-OpenTP1-identifier*\lib is added at the beginning.
PATH	*OpenTP1-setup-folder-corresponding-to-OpenTP1-identifier*\bin is added at the beginning in order to execute the corresponding OpenTP1's commands with top priority.

Command arguments

- *identifier* ~<1 to 4 single-byte alphanumeric characters>

  Specifies the character string that was specified when the multiOpenTP1 was set up.

  If you execute the command without specifying a command argument, the command creates a console for manipulating the original OpenTP1.

Output messages

Message ID	Description	Output destination
KFCA26551-E	Command execution failed.	Standard error
KFCA26552-E	A specified command argument is invalid.	Standard error
KFCA26555-I	Help message	Standard error

Notes

- When you use a multiOpenTP1, make sure that you use the console created by this command to change environment variables. If you use any other method to change environment variables, the operation cannot be guaranteed.
- If you execute the `dcsetupml` command, all installed program products are copied to the multiOpenTP1. Use the console created by this command to manipulate other program products as well, in addition to TP1/Server Base (such as TP1/FS/Direct Access, TP1/FS/Table Access, and TP1/Message Control).
- If you use **Add or Remove Programs** from **Control Panel** to uninstall the original OpenTP1 in the multiOpenTP1 system, all multiOpenTP1s will also be deleted. In such a case, all folders under the multiOpenTP1 setup folders (`bin`, `etc`, `examples`, `include`, `lib`, `spool`, `tmp`, and `tools`) are also deleted. If you have created files under these folders, be sure to back them up before you start uninstallation of the original OpenTP1.
- If a serious failure occurs during execution of this command, back up the following data as soon as possible and then contact maintenance personnel:
  - `%TEMP%\dcsetupml.log`
  - `%TEMP%\trnlnkrm.log`
  - `%DCDIR%\spool\errlog\*.*` (`%DCDIR%` indicates the original OpenTP1's installation folder)
- If you use this command to create a console for a multiOpenTP1 and then change the multiOpenTP1 definitions or the `DCCONFPATH` and `DCUAPCONFPATH` environment variable settings, you must execute this command again after the change to re-create the console for the multiOpenTP1.

dcsetupml (sets up a multiOpenTP1 or cancels setup of a multiOpenTP1)

Format

```
dcsetupml [{-u|-d} ]
          multiOpenTP1-setup-directory  identifier
```

Function

This command sets up (or deletes) a multiOpenTP1 in a specified folder. The OpenTP1 identifier is used to identify each OpenTP1 within the same node. The user specifies this identifier as 1 to 4 single-byte alphanumeric characters. This command can be executed only when OpenTP1 is offline.

You can execute this command only from the console for the original OpenTP1.

Options

- `-u`

  Updates the multiOpenTP1 whose identifier is specified in the command argument to its most recent status.

- `-d`

  Deletes the multiOpenTP1 whose identifier is specified in the command argument.

Command arguments

- *multiOpenTP1-setup-directory* ~<path name>

  Specifies the folder in which the multiOpenTP1 is to be set up, expressed as 1 to 50 characters.

- *identifier* ~<1 to 4 single-byte alphanumeric characters>

  Specifies the character string used to identify the multiOpenTP1.

Output messages

Message ID	Description	Output destination
KFCA26551-E	Command execution failed.	Standard error
KFCA26552-E	A specified command argument is invalid.	Standard error
KFCA26553-I	Help message	Standard output, standard error

Notes

- Execute this command while the original OpenTP1 and the target multiOpenTP1 are not running.
- This command must be executed by a user with Administrators group permissions.
- This command copies the folders contained in the OpenTP1 installation folder (`bin`, `etc`, `examples`, `include`, `lib`, and `tools`). Note that if you have changed or added files in folders under the OpenTP1 installation folder, those files are also copied as is.
- When you execute this command, all installed program products are copied to the multiOpenTP1. Use the console created by the `dcmakecon` command to manipulate other program products as well, in addition to TP1/Server Base (such as TP1/FS/Direct Access, TP1/FS/Table Access, and TP1/Message Control).
- If you intend to execute this command with the `-u` or `-d` option specified and have created user-specific files in the folders under the multiOpenTP1 setup folder (`bin`, `etc`, `examples`, `include`, `lib`, `spool`, `tmp`, and `tools`), back up these files before you execute the command.
- If you have connected another company's resource manager to the multiOpenTP1 and you uninstall TP1/Server Base or update the multiOpenTP1, the resource manager information will be reset. When you update or set up the multiOpenTP1 again, use the `trnlnkrm` command again to connect the other company's resource manager.
- If you execute this command or perform uninstallation while another OpenTP1 command is executing or a command prompt is open under the OpenTP1 setup folder, folders and files might remain, or copy processing might fail. Make sure that you execute this command or perform uninstallation only after all other command executions and folder operations have been completed.
- In a system of multiOpenTP1s, you must set a unique port number for the name service of each OpenTP1. Also make sure that all other port numbers that are specified explicitly are unique. If any port number is duplicated, the `KFCA00107-E` message will be displayed in Event Viewer and OpenTP1 might terminate abnormally.
- Do not close the console or press the **Ctrl** + **Break** keys together while this command is executing.
- If a serious failure occurs during execution of this command, back up the following data as soon as possible and then contact maintenance personnel:
  - `%TEMP%\dcsetupml.log`
  - `%TEMP%\trnlnkrm.log`

- `%DCDIR%\spool\errlog\*.*` (`%DCDIR%` indicates the original OpenTP1's installation folder)

njsmkdll (creates a transaction control DLL)

Format

```
njsmkdll -d transaction-control-object-name
         [-R OpenTP1-provided-RM-name[,OpenTP1-provided-RM-name]...]
         [-r
RM-name-that-is-not-provided-by-OpenTP1[,RM-name-that-is-not-provided-by-OpenTP1]...]
         [-o RM-related-object-name[ΔRM-related-object-name]...]
```

Function

If you use TP1/Extension for .NET Framework to link resource managers and the XA interface, you must use this command to create a DLL for controlling transactions. A manifest file is embedded in the created DLL.

Options

- `-d` *transaction-control-object-name* ~<1 to 12 alphanumeric characters>

  Specifies a name for the transaction control DLL. When you execute this command, a file named *transaction-control-DLL-name*`.dll` is created under the `%DCDIR%\spool\trnrmcmd\userdll` folder.

- `-R` *OpenTP1-provided-RM-name* ~<1 to 32 alphanumeric characters>

  Specifies the name of an OpenTP1-provided resource manager that is accessed from the UAP. A resource manager that has not been registered in OpenTP1 cannot be specified. The following OpenTP1-provided resource manager names can be specified:

  - `OpenTP1_TAM`: Resource manager name of TP1/FS/Table Access
  - `OpenTP1_MCF`: Resource manager name of TP1/Message Control

  If you specify multiple resource manager names, delimit them with a comma (`,`).

- `-r` *RM-name-that-is-not-provided-by-OpenTP1* ~<1 to 32 alphanumeric characters>

  Specifies the name of a resource manager that is accessed from the UAP and that is not provided by OpenTP1. A resource manager that has not been registered in OpenTP1 cannot be specified. If you specify multiple resource manager names, delimit them with a comma (`,`).

- `-o` *RM-related-object-name* ~<absolute path name>

  Specifies the absolute path (MS-DOS format file containing no spaces) of an object file (object file for the XA interface) that is related to a resource manager. If you specify multiple resource manager-related object names, delimit them with a space.

Output messages

Message ID	Description	Output destination
KFCA32298-I	Help message	Standard output

Notes

The `trnmkobj` command is executed within this command. For details about the `trnmkobj` command, see *trnmkobj (createsa transaction control object file)*.

ntbstart (starts OpenTP1)

Format

```
ntbstart [-n][-U]
```

Function

This command starts the OpenTP1 service that corresponds to the DCDIR environment variable in the command execution environment.

Options

- -n

  Forcibly starts the OpenTP1 normally regardless of the previous termination mode. If this option is omitted, the command selects the startup mode based on the previous termination mode:

  - If the previous termination mode was normal termination: Normal startup
  - If the previous termination mode was not normal termination: Restart

  If the OpenTP1 service is to be started normally, this option is ignored.

- -U

  If the startup mode is restart, this option suppresses startup of the user server.

  You use this option when there is no need to start the user server, such as when a failure has occurred and OpenTP1 is to be run in online mode in order to recover the OpenTP1 status.

Output messages

Message ID	Description	Output destination
KFCA26560-I	OpenTP1 is starting.	Standard output
KFCA26561-I	OpenTP1 has started.	Standard output
KFCA26562-E	A failure occurred during OpenTP1 start processing.	Standard error
KFCA26563-E	A specified option is invalid.	Standard error
KFCA26564-E	A combination of options is invalid.	Standard error
KFCA26565-E	A specified command argument is invalid.	Standard error
KFCA26567-E	Memory allocation failed.	Standard error
KFCA26569-E	The OpenTP1 service has not been registered.	Standard error

Message ID	Description	Output destination
KFCA26570-E	The user does not have permission to start the OpenTP1 service.	Standard error
KFCA26571-E	Connection to the service control manager failed.	Standard error
KFCA26572-E	Acquisition of the OpenTP1 service status failed.	Standard error
KFCA26573-E	The OpenTP1 service is not stopped.	Standard error
KFCA26574-E	The OpenTP1 service is disabled.	Standard error
KFCA26575-E	Access to registry failed.	Standard error
KFCA26578-I	Help message	Standard error

Notes

Note the following about executing this command with the -U option specified:

- Use the dcsvstart command if you wish to start the user server after online operation has started.
- If you wish to recover the user server's configuration after online operation has started, terminate the system in a mode other than normal termination or forced normal termination and then start online operation by executing this command with the -U option omitted.
- If, during the previous online session, you executed this command with the -U option specified during a rerun after you had used the remote API facility, it will not be possible to recover the remote API's configuration. If you use the remote API facility after executing the command with the -U option specified, take one of the following actions:
  - After online operation has started, use the dcsvstart command to manually start the RAP-processing listener.
  - Terminate the system in a mode other than normal termination or forced normal termination and then start online operation by executing this command with the -U option omitted.
- If you use the server shutdown inheriting facility, the following problems arise:
  - The schedule shutdown status is not inherited if online operation is started by a rerun by executing this command with the -U option specified.
  - If the conditions listed below are satisfied, the schedule shutdown status of online operation in existence before this command was executed with the -U option specified is passed to SPP:
    - There is an SPP in schedule shutdown status.

- The online operation started by a rerun by executing this command with the -U option specified was terminated by planned or forced termination and then this command was used again to start the OpenTP1.

  To avoid these problems, you must take the following actions:

  - Perform normal termination to terminate the online operation started by a rerun by executing this command with the -U option specified or perform forced normal start.
  - Specify hold_recovery=N in all user service definitions.
  - Specify scd_hold_recovery_count=0 in the schedule service definition.

- If you use the system switchover facility, a standby OpenTP1 ignores the -n option.

ntbtail (outputs the contents of a specified file to the console)

Format

```
ntbtail [-i update-check-interval] -t p
ntbtail [-i update-check-interval] redirect-file-name
```

Function

This command reads a specified file at a specified interval and outputs the updated contents to the console. If the specified file has managed generations, the command reads the contents of the file with generation number 1 or 2 at the end of the file name and then outputs those contents to the console.

Options

- `-i` *update-check-interval* ~<unsigned integer>((1 to 60))<<1>>(seconds)

  Specifies the interval at which the file is to be checked for updated contents.

- `-t p`

  Specifies that the contents of the standard output redirect file are to be read and output to the console. This option provides the same functionality as the `tp1console` command. For details about the `tp1console` command, see *tp1console (outputs the standard output redirect file to the console)*.

Command arguments

- *redirect-file-name* ~<path name consisting of 1 to 260 characters>

  Specifies the name of the file whose updated file contents are to be output.

  Generations of the specified file must be managed according to the following conditions:

  - The file name does not have an extension.
  - The name of a file that has two managed generations must end with generation number 1 or 2.
  - Neither the current file nor the standby file is deleted during generation swapping.

  A file with managed generations that is output by this command cannot be deleted until this command has terminated.

  For the file name, specify the name without the generation number at the end. The following example outputs the files `%DCDIR%\spool\prclog1` and `%DCDIR%\spool\prclog2`:

```
ntbtail %DCDIR%\spool\prclog
```

Output messages

Message ID	Description	Output destination
KFCA26526-E	Command execution failed.	Standard error
KFCA26527-I	Help message	Standard output

Notes

- Make sure that this command has terminated before you set up OpenTP1 (install or uninstall OpenTP1 or set up a multiOpenTP1).
- The `dc_adm_call_command` function cannot be used to execute this command from a UAP. This command cannot be specified in the `user_command` or the `user_command_online` operand in the system environment definition.
- This command determines the file generation to be output on the basis of the modification dates of the file specified in the argument. If you update data in the current file specified in the argument, use the `_commit()` or `FlushFileBuffers()` function or close the file to write data to disk and set the most recent file modification date. If the current file's modification date is not up-to-date, the correct current file cannot be selected, in which case the command might not function as intended.
- The contents of the specified file before this command was executed are not output to the console. The command outputs sequentially to the console the contents written after command execution has begun.
- If the file specified in *redirect-file-name* has managed generations and the file generations are swapped, the file containing the most recent generation after swapping is the one that is subject to output by this command.
- If you use a multiOpenTP1 system, execute this command from a command prompt at the console created by the `dcmakecon` command. For details, see *dcmakecon (creates a multiOpenTP1 console)*.
- To terminate this command, press the **Ctrl** and **C** keys simultaneously at the console or perform console close processing.

tp1console (outputs the standard output redirect file to the console)

Format

```
tp1console
```

Function

This command reads the standard output redirect file at a specific interval and outputs the updated contents to the console.

This command can be executed only when `Y` is specified in the `redirect_file` operand in the system environment definition. For details about the `redirect_file` operand, see *System environment definition* in *5.3 Details of system definitions*. For details about output of the standard output redirect file to the console, see *2.2.2 Console output facility for standard output redirect files*.

Output messages

Message ID	Description	Output destination
KFCA26526-E	Command execution failed.	Standard error
KFCA26527-I	Help message	Standard output

Notes

- Make sure that this command has terminated before you set up OpenTP1 (install or uninstall OpenTP1 or set up a multiOpenTP1).
- The `dc_adm_call_command` function cannot be used to execute this command from a UAP. This command cannot be specified in the `user_command` or the `user_command_online` operand in the system environment definition.
- The standard output redirect file to be output by this command cannot be deleted until this command has terminated.
- The contents of the standard output redirect file before this command was executed are not output to the console. The command outputs sequentially to the console the contents written after command execution has begun.
- If the standard output redirect file has managed generations and the file generations are swapped, the file containing the most recent generation after swapping is the one that is subject to output by this command.
- If you use the original OpenTP1, you can execute this command from the Windows **Start** menu. To start the command from the Desktop, choose **Start**, **Programs**, **OpenTP1**, then **OpenTP1 Console**.

- If you execute this command on a multiOpenTP1 system, execute it from the command prompt at the console that was created by the `dcmakecon` command. For details, see *dcmakecon (creates a multiOpenTP1 console)*.
- To terminate this command, press the **Ctrl** and **C** keys simultaneously at the console or perform console close processing.
- If a failure occurs during execution of this command, the `KFCA26526-E` message is issued and the system is placed in `pause` status. Press the return key or perform console close processing to release the `pause` status.
- This command is provided as a batch file. Do not change the contents of this batch file. If you change the contents of the batch file and then execute it, operation cannot be guaranteed.

Appendix

A. Lists of OpenTP1 Processes

A. Lists of OpenTP1 Processes

This appendix provides lists by product of the system service processes used by OpenTP1 for Windows.

Note

The column *Number of system service processes* indicates the number of system service processes per node when a single OpenTP1 is run per node.

(1) TP1/Server Base

The following table lists and describes the system service processes of TP1/Server Base.

Table A-1: System service processes (TP1/Server Base)

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
admrsvre	0 or more	Partial recovery	Process service definition	None
cltcond	0 or more	CUP execution service	Client service definition	None
cltd	0 or 1	Client extended service	Client service definition	None
clttrnd	0 or more	Client execution process	Client service definition	None
cpdd	1 or more	Checkpoint dump service	Journal service definition Checkpoint dump service definition	Checkpoint dump file
dcservice	1	Windows service process	None	None
itvd	1	Interval timer service	Interval service definition	None
jnld	1	Journal management service	Journal service definition	None
jnliod	1 to 16[#1]	Journal file input/ output service	None	System journal file
jnlswd	1	Journal file management service	System journal service definition	None
jnlutld	0 or 1[#2]	Journal utility service	System journal service definition	System journal file

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
logd	1	Log service	Log service definition	Message log file
namaudtd	0 or 1	Name service	Name service definition	None
namd	1[#3]	Name service	Name service definition	None
prcd	1	Process service (superuser process)	Process service definition	None
prctee	0 or 1[#4]	Standard output redirect service	System environment definition	Standard output redirect file
prfiop[#5]	0, 8, or 9[#6]	Performance verification trace acquisition service	System common definition	Trace file
qued	0 or 1	Queue service	MCF-message queue service definition	MCF-message queue file
rapclman	0 or 1	Report API facility	RAP-processing client manager service definition	None
raplisnr	0 to 1024	Report API facility	RAP-processing listener service definition	Trace file
rapservr	0 to 1024	Report API facility	RAP-processing listener service definition	Trace file
rtsspp	0 or more	Real-time statistics service	Real-time acquisition item definition	RTS log file
rtssup	0 or more	Real-time statistics service	Real-time acquisition item definition	RTS log file
scdd	1[#3]	Scheduler	Schedule service definition	None
scdmltd	0 or more	Scheduler	Schedule service definition	None
stsd	1	Status service	Status service definition	Status file
tjld	1	Transaction journal service	None	Transaction recovery journal file
trnd	1[#3]	Transaction management service	Transaction service definition	None
trnrmd	1	Resource manager monitoring service	Transaction service definition	None

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
trnrvd	1 or more	Transaction recovery service	Transaction service definition	None

#1

This value is obtained from the following formula:

a x *b*

a: 2 when dual journals are used; 1 otherwise.

b: Maximum number of dispersions for the parallel access facility for system journal files.

#2

If you specify Y in the jnl_auto_unload operand in the system journal service definition, the number of system service processes is 1.

#3

If you monitor OpenTP1 processes, specify one of these processes as the process to be monitored.

#4

If you specify Y in the redirect_file operand in the system environment definition, the number of system service processes is 1.

#5

This is a process for acquiring traces; it is not displayed by the prcls command. To check whether the process is running, use Windows Task Manager's **Processes** page.

#6

The number of system service processes is as follows:

- When N is specified in the prf_trace operand in the system common definition

  The number of system service processes is 0.

- When Y is specified in the prf_trace operand in the system common definition

  If MCF is not used, the number of system service processes is 8.

If MCF is used, the number of system service processes is 9.

(2) TP1/FS/Direct Access

The following table lists and describes the system service processes of TP1/FS/Direct Access.

Table A-2: System service processes (TP1/FS/Direct Access)

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
damd	0 or 1	DAM service	DAM service definition	DAM file

(3) TP1/FS/Table Access

The following table lists and describes the system service processes of TP1/FS/Table Access.

Table A-3: System service processes (TP1/FS/Table Access)

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
tamd	0 or 1	TAM service	TAM service definition	TAM file
tamiod	0 or 1	TAM file input/output service	TAM service definition	TAM file

(4) TP1/Message Control

The following table lists and describes the system service processes of TP1/Message Control.

Table A-4: System service processes (TP1/Message Control)

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
mapsmgrd	0 or 1	Mapping service	Mapping service definitions Mapping service attribute definitions	None
mcfmngrd	0 or 1	MCF manager	MCF manager definition	Trace file

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
User-specified name#	0 or more	MCF communication service	MCF communication configuration definition MCF application definition	None

#

If TP1/Messaging is used, this must be `mcfutcpd` and `mcfupsvd`.

(5) TP1/Message Queue

The following table lists and describes the system service processes of TP1/Message Queue.

Table A-5: System service processes (TP1/Message Queue)

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
`mqacmd`	0 or 1	Message queuing (TP1/Message Queue) on-line command service	MQA service definition	MQA queue file
`mqad`	0 or 1	Message queuing (TP1/Message Queue) MQA service	MQA service definition	MQA queue file MQ management information file
`mqaiod`	0 or 1 or more	Message queuing (TP1/Message Queue) queue file input/output service	MQA service definition	MQA queue file
`mqamnd`	0 or 1	Message queuing (TP1/Message Queue) expiration message monitoring service	MQA service definition	None
`mqcdtcp`	0 or 1	Message queuing (TP1/Message Queue) MQC service	MQA service definition MQC service definition	Trace file
`mqcgwp`	0 or 1 or more	Message queuing (TP1/Message Queue) MQC gateway server	User service definition	MQA queue file Trace file

Executable file name	Number of system service processes	Service	Related system definitions	Files to be input or output
mqrspp	0 or 1 or more	Message queuing (TP1/Message Queue) repository management server SPP	User service definition (%DCCONFPATH%\mqrspp)	MQA queue file
mqrsup	0 or 1	Message queuing (TP1/Message Queue) repository management server SUP	User service definition (%DCCONFPATH%\mqrsup)	MQA queue file Trace file
mqtdtcp	0 or 1 or more	Message queuing (TP1/Message Queue) MQT communication service	Definition of the MQT communication configuration MQT service definition	MQA queue file Channel management information storage file Trace file User definition file MQ management information file
mqtmngd	0 or 1	Message queuing (TP1/Message Queue) MQT communication manager service	MQA service definition	Channel management information storage file

Index

A

adm_message_option
- user service default definition 137
- user service definition 143

all_node (system common definition) 97
all_node_ex (system common definition) 99
all_node_ex_extend_number (system common definition) 99
all_node_extend_number (system common definition) 99
alternate sending
- ending (mcftedalt) 227
- starting (mcftstalt) 227

application
- displaying status of (mcfalsap) 225
- releasing, from shutdown status (mcfaactap) 225
- shutting down (mcfadctap) 225

application abnormal termination count, initializing (mcfaclcap) 226
application program, starting (mcfuevt) 226
application timer start request
- deleting (mcfadltap) 226
- displaying (mcfalstap) 226

archive status, displaying (jnlarls) 220
assurance_count (checkpoint dump service definition) 110
atomic_update
- user service default definition 135
- user service definition 141

audit log facility, setting up environment for (dcauditsetup) 218
auto_restart
- user service default definition 135
- user service definition 142

automatic unloading facility, controling (jnlatunl) 220

B

balance_count
- client service definition 118
- user service default definition 135
- user service definition 141

buffer group, displaying usage status of (mcftlsbuf) 227

C

C runtime library, installing 33
cache blocks, number of
- acquiring (damchinf) 222
- setting threshold for (damchdef) 222

cache, clearing (namalivechk) 217
cancel_normal_terminate
- user service default definition 136
- user service definition 142

client service definition 160
client_uid_check (system common definition) 98
clt_conf (system service configuration definition) 96
clt_cup_conf (client service definition) 119
clt_inquire_time (client service definition) 119
clt_port (client service definition) 119
clt_trn_conf (client service definition) 119
cltcon_port (client service definition) 119
clttrn_port (client service definition) 119
COBOL2002 Development Manager settings 39
code-level functions, list of 20
command
- specifying (COBOL2002) 36
- specifying (Net Express) 40

command, details of
- common to OpenTP1 for Windows and OpenTP1 for UNIX 231
- specific to OpenTP1 for Windows 248

concurrent message processing status, displaying (mcftlstrd) 225
connection

changing (mcftchcn) 225
displaying status of (mcftlscn) 225
establishing (mcftactcn) 225
releasing (mcftdctcn) 225
console output facility for standard output redirect files 15
console_output (system environment definition) 95
core_shm_suppress
user service default definition 139
user service definition 145
core_suppress_watch_time (system common definition) 98
critical
user service default definition 135
user service definition 142
cup_balance_count (client service definition) 119
cup_parallel_count (client service definition) 119

D

dam_added_file (DAM service definition) 116
dam_cache_attribute (DAM service definition) 116
dam_cache_reuse_from (DAM service definition) 117
dam_cache_size (DAM service definition) 116
dam_cache_size_fix (DAM service definition) 116
dam_conf (system service configuration definition) 96
dam_default_cache_num (DAM service definition) 117
dam_ex_refer_read (DAM service definition) 117
dam_io_error_occur (DAM service definition) 117
dam_io_interval (DAM service definition) 117
dam_kb_size (DAM service definition) 117
dam_max_block_size (DAM service definition) 117
dam_message_level (DAM service definition) 116
dam_tran_process_count (DAM service definition) 116
dam_transaction_access (DAM service definition) 117
dam_update_block (DAM service definition) 116
dam_update_block_over (DAM service definition) 116
damadd (adds a logical file) 222
dambkup (backs up a physical file) 222
damcache (DAM service definition) 117
damchdef (sets a threshold for the number of cache blocks) 222
damchinf (acquires the number of cache blocks) 222
damchlmt (DAM service definition) 117
damdel (deletes a physical file) 222
damfile (DAM service definition) 117
damfrc (restores logical files) 222
damhold (logically shuts down a logical file) 222
damload (initializes a physical file) 222
damls (displays the status of logical files) 222
damrles (releases a logical file from shutdown status) 222
damrm (detaches a logical file) 222
damrstr (restores a physical file) 222
DCADMDEBUG (system environment definition) 96
dcauditsetup (sets up the environment for the audit log facility) 218
dcbindht (system common definition) 100
DCCONFPATH (system environment definition)
dcputenv 96
putenv 95
dccspool (deletes troubleshooting information) 214
dcdefchk (checks system definitions) 214
dcdls (checks the setup status of multiOpenTP1s) 228
DCFPL_CONNECT_RETRY_COUNT (user service definition) 146
DCFPL_CONNECT_RETRY_INTERVAL (user service definition) 146
dcjchconf (specifies a system definition operand) 213
dcjcmdex (execute an OpenTP1 command from a scenario template) 213
dcjnamch (updates a domain definition file) 214
dcmakecon (creates a multiOpenTP1 console) 228
dcmakeup (reserves and releases resources for OpenTP1 internal control) 213
dcmapchg (changes the path name of a map file) 227
dcmapls (displays the loaded resources of a map file) 227
dcmarea (multi-node structure definition) 114
dcmstart (starts a multi-node area or subarea) 213
dcmstart_watch_time (multi-node structure definition) 114

dcmstop (terminates a multi-node area or subarea) 213
dcmstop_watch_time (multi-node structure definition) 114
dcndls (displays the status of one or more OpenTP1 nodes) 214
dcpplist (displays product information) 215
dcprcport (multi-node physical definition) 115
dcrasget (acquires maintenance data) 214
dcreport (edits and outputs system statistical information to standard output in real time) 214
dcreset (restarts the process service and updates it from the definitions) 213
dcsetup (registers OpenTP1 into the OS or deletes OpenTP1 from the OS) 213
dcsetupml (sets up a multiOpenTP1 or cancels setup of a multiOpenTP1) 228
dcshmls (displays the status of shared memory) 214
dcstart (starts OpenTP1) 213
dcstart_wakeup_retry_count (system common definition) 100
dcstart_wakeup_retry_interval (system common definition) 100
dcstats (starts and terminates acquisition of system statistical information) 213
dcstop (terminates OpenTP1) 213
dcsvgdef (user service network definition) 123
dcsvstart
    starts one or more servers 215
    system service configuration definition 97
    user service structure definition 97
dcsvstop (terminates one or more servers) 215
DCSYSLOGOUT (log service definition) 114
DCUAPCONFPATH (system environment definition)
    dcputenv 96
    putenv 96
deadlock information file, deleting (lckrminf) 217
deadlock_priority
    user service default definition 134
    user service definition 141
definition file, creating 58
Detailed Information dialog box 201
domain configuration
    changing (using domain definition file) (namchgfl) 217
    changing (using system common definition) (namndchg) 217
domain definition file, updating (dcjnamch) 214
domain representative schedule service
    deleting (namdomainsetup) 217
    registering (namdomainsetup) 217
domain_masters_addr (system common definition) 98
domain_masters_port (system common definition) 98
domain_use_dns (system common definition) 98
dynamic_shmpool_size (system environment definition) 95

E

environment setup
    procedure 54
    when Visual Studio is also used (version earlier than COBOL2002 02-00) 40
    with GUI 195
environment variable settings 60
error
    cause 190
    event 190

F

fil_prf_trace_delay_time (system common definition) 100
fil_prf_trace_option (system common definition) 100
filbkup (backs up an OpenTP1 file system) 218
filchgrp (changes an OpenTP1 file group) 218
filchmod (changes the access permission mode for an OpenTP1 file) 218
filchown (changes an OpenTP1 file owner) 218
fills (displays the contents of an OpenTP1 file system) 218
filmkfs (initializes an OpenTP1 file system) 218
filrstr (restores an OpenTP1 file system) 218
filstatfs (displays the status of an OpenTP1 file system) 218
functions
    description of 7

differences between OpenTP1 for Windows and OpenTP1 for UNIX 8, 20
for checking standard error 13
for checking standard output 13
list of 8

G

global archive unload journal file
editing and outputting (jnledit) 221
outputting record from (jnlrput) 221
sorting and merging, in time order (jnlsort) 221
groups
transaction service definition 106
user service default definition 138
user service definition 145

H

ha_conf (system service configuration definition) 96
hold
user service default definition 134
user service definition 141
hold_recovery
user service default definition 134
user service definition 141

I

input/output queue content, copying (mcftdmpqu) 228
ipc_backlog_count
system common definition 99
user service default definition 138
user service definition 145
ipc_conn_interval
system common definition 99
user service default definition 138
user service definition 144
ipc_header_recv_time
system common definition 99
user service default definition 138
user service definition 144
ipc_listen_sockbufset
system common definition 99
user service default definition 138
user service definition 144
ipc_recvbuf_size
system common definition 99
user service default definition 138
user service definition 144
ipc_send_count
system common definition 99
user service default definition 138
user service definition 144
ipc_send_interval
system common definition 99
user service default definition 138
user service definition 144
ipc_sendbuf_size
system common definition 99
user service default definition 138
user service definition 144
ipc_sockctl_highwater
RAP-processing listener service definition 126
system common definition 98
user service default definition 138
user service definition 144
ipc_sockctl_watchtime
RAP-processing listener service definition 126
system common definition 98
user service default definition 138
user service definition 144
ipc_tcpnodelay
schedule service definition 103
user service default definition 139
user service definition 145
ist_conf (system service configuration definition) 96
ist_node (IST service definition) 120
ist_node_group (IST service definition) 120
istdef (IST service definition) 120

J

jar_conf (system service configuration definition) 96
jnl_arc_buff_size (system journal service definition) 109

jnl_arc_check_level (system journal service definition) 110
jnl_arc_ipc_buff_size
    global archive journal service definition 115
    journal service definition 108
jnl_arc_max_datasize
    archive journal service definition 116
    system journal service definition 109
jnl_arc_name (system journal service definition) 109
jnl_arc_rec_kind (system journal service definition) 109
jnl_arc_terminate_check (system journal service definition) 109
jnl_arc_terminate_timeout
    global archive journal service definition 115
    journal service definition 108
jnl_arc_trn_stat (system journal service definition) 110
jnl_arc_uj_code (system journal service definition) 109
jnl_auto_unload (system journal service definition) 110
jnl_auto_unload_path (system journal service definition) 110
jnl_cdinterval (system journal service definition) 109
jnl_dual
    archive journal service definition 115
    checkpoint dump service definition 111
    system journal service definition 109
jnl_fileless_option (system common definition) 100
jnl_max_datasize
    checkpoint dump service definition 110
    system journal service definition 109
jnl_max_file_dispersion
    archive journal service definition 116
    system journal service definition 110
jnl_min_file_dispersion
    archive journal service definition 116
    system journal service definition 110
jnl_objservername (checkpoint dump service definition) 110
jnl_prf_event_trace_level (system common definition) 100
jnl_reduced_mode (checkpoint dump service definition) 110
jnl_rerun_reserved_file_open (system journal service definition) 109
jnl_rerun_swap
    archive journal service definition 115
    system journal service definition 109
jnl_reserved_file_auto_open (checkpoint dump service definition) 111
jnl_singleoperation
    archive journal service definition 115
    checkpoint dump service definition 111
    system journal service definition 109
jnl_tran_optimum_level (journal service definition) 108
jnl_unload_check
    archive journal service definition 116
    system journal service definition 110
jnl_watch_time (journal service definition) 109
jnladdfg
    archive journal service definition 116
    checkpoint dump service definition 111
    system journal service definition 110
jnladdpf
    allocates a journal physical file 219
    archive journal service definition 116
    checkpoint dump service definition 111
    system journal service definition 110
jnlardis (forcibly releases resource group connection) 222
jnlarls (displays archive status) 220
jnlatunl (controls the automatic unloading facility) 220
jnlchgfg (changes the status of journal files) 220
jnlclsfg (closes journal files) 219
jnlcolc (collects journals required for file recovery) 220
jnlcopy (copies unload journal files) 220
jnldelpf (deletes a journal physical file) 219
jnldfsv
    global archive journal service definition 115
    journal service definition 109
jnledit (edits and outputs unload journal files or global archive unload journal files) 221

jnlinit (initializes a journal file) 219
jnlls (displays journal file information) 219
jnlmcst (outputs MCF statistics) 221
jnlmkrf (recovers journal files) 220
jnlopnfg (opens journal files) 219
jnlrinf (displays journal file information that was read during restart) 219
jnlrm (deletes a journal file) 219
jnlrput (outputs records from unload journal files or global archive unload journal files) 221
jnlsort (sorts and merges unload journal files and global archive unload journal files in time order) 221
jnlstts (outputs statistical information) 221
jnlswpfg (swaps journal files) 219
jnlunlfg (unloads journal files) 220
journal file
    changing status of (jnlchgfg) 220
    closing (jnlclsfg) 219
    deleting (jnlrm) 219
    initializing (jnlinit) 219
    opening (jnlopnfg) 219
    recovering (jnlmkrf) 220
    swapping (jnlswpfg) 219
    unloading (jnlunlfg) 220
journal file information
    displaying (jnlls) 219
    that was read during restart, displaying (jnlrinf) 219
journal physical file
    allocating (jnladdpf) 219
    deleting (jnldelpf) 219
journal required for file recovery, collecting (jnlcolc) 220
journal service definition 159
jp1_use (system common definition) 98

L

LANG (system common definition) 100
lck_deadlock_info (lock service definition) 101
lck_deadlock_info_remove (lock service definition) 101
lck_deadlock_info_remove_level (lock service definition) 101
lck_limit_fordam (lock service definition) 100
lck_limit_formqa (lock service definition) 101
lck_limit_fortam (lock service definition) 100
lck_limit_foruser (lock service definition) 100
lck_prf_trace_level (lock service definition) 101
lck_release_detect (lock service definition) 101
lck_release_detect_interval (lock service definition) 101
lck_wait_priority
    user service default definition 135
    user service definition 142
lck_wait_timeout (lock service definition) 101
lckls (displays lock information) 217
lckpool (displays pool information from the lock control table) 217
lckrminf (deletes deadlock and timeout information files) 217
leveldown_queue_count
    user service default definition 138
    user service definition 144
levelup_queue_count
    user service default definition 138
    user service definition 144
library search path 49
linking
    with HiRDB 68
    with Oracle 73
    with SQL Server 81
linking databases, settings for 68
loadcheck_interval
    user service default definition 138
    user service definition 144
loadlevel_message
    user service default definition 138
    user service definition 145
lock information, displaying (lckls) 217
Log File dialog box 204
log_audit_count (log service definition) 113
log_audit_message
    log service definition 113
    RAP-processing client manager service definition 126
    RAP-processing listener service definition 126
    user service default definition 139

user service definition 146
log_audit_out (log service definition) 113
log_audit_out_suppress
RAP-processing client manager service definition 126
RAP-processing listener service definition 126
user service default definition 139
user service definition 146
log_audit_path (log service definition) 113
log_audit_size (log service definition) 113
log_filesize (log service definition) 111
log_jerr_rint (log service definition) 113
log_jp1_allno (log service definition) 112
log_jp1_date (log service definition) 112
log_jp1_hostname (log service definition) 112
log_jp1_pgmid (log service definition) 112
log_jp1_prcid (log service definition) 112
log_jp1_prcno (log service definition) 112
log_jp1_sysid (log service definition) 112
log_jp1_time (log service definition) 112
log_msg_allno (log service definition) 111
log_msg_console (log service definition) 111
log_msg_date (log service definition) 111
log_msg_hostname (log service definition) 112
log_msg_pgmid (log service definition) 112
log_msg_prcid (log service definition) 111
log_msg_prcno (log service definition) 111
log_msg_sysid (log service definition) 111
log_msg_time (log service definition) 112
log_netm_allno (log service definition) 112
log_netm_date (log service definition) 112
log_netm_hostname (log service definition) 112
log_netm_out (log service definition) 112
log_netm_pgmid (log service definition) 112
log_netm_prcid (log service definition) 112
log_netm_prcno (log service definition) 112
log_netm_sysid (log service definition) 112
log_netm_time (log service definition) 112
log_notify_allno (log service definition) 112
log_notify_date (log service definition) 113
log_notify_hostname (log service definition) 113
log_notify_out (log service definition) 112
log_notify_pgmid (log service definition) 113
log_notify_prcid (log service definition) 112
log_notify_prcno (log service definition) 112
log_notify_sysid (log service definition) 112
log_notify_time (log service definition) 113
log_syslog_allno (log service definition) 113
log_syslog_append_nodeid (log service definition) 113
log_syslog_date (log service definition) 113
log_syslog_elist (log service definition) 113
log_syslog_elist_rint (log service definition) 113
log_syslog_hostname (log service definition) 113
log_syslog_out (log service definition) 113
log_syslog_pgmid (log service definition) 113
log_syslog_prcid (log service definition) 113
log_syslog_prcno (log service definition) 113
log_syslog_synchro (log service definition) 113
log_syslog_sysid (log service definition) 113
log_syslog_time (log service definition) 113
logcat (displays the contents of a message log file) 218
logcon (changes the setting for the real-time output facility for message logs) 218
logical file
adding (damadd) 222
detaching (damrm) 222
displaying status of (damls) 222
logically shutting down (damhold) 222
releasing from shutdown status (damrles) 222
restoring (damfrc) 222
logical terminal
deleting output queues for (mcftdlqle) 226
displaying status of (mcftlsle) 226
forcibly ending continuous inquiry-response processing for (mcftendct) 227
from hold status, releasing output queue processing for (mcftrlsoq) 226
in hold status, placing output queue processing for (mcfthldoq) 226
releasing, from shutdown status (mcftactle) 226
shutting down (mcftdctle) 226
skipping first message in message queue for (mcftspqle) 226
logical terminal-related message journal

ending acquisition of (mcftdctmj) 226
starting acquisition of (mcftactmj) 226

M

maintenance data, acquiring (dcrasget) 214
make_queue_on_starting
user service default definition 137
user service definition 144
makefile
example (C language: MHP) 34
example (C language: SPP) 33
example (COBOL2002: MHP) 38
example (COBOL2002: SPP) 37
example (Net Express: MHP) 42
example (Net Express: SPP) 40
manifest file, preparing 33
manual
how to use this 3
relationship to other OpenTP1 series manuals 2
map file path name, changing (dcmapchg) 227
map file, displaying loaded resources of (dcmapls) 227
max_open_fds
system service common information definition 149
user service default definition 136
user service definition 143
XATMI communication service definition 122
max_socket_descriptors
global archive journal service definition 115
journal service definition 108
name service definition 102
process service definition 102
RAP-processing listener service definition 125
schedule service definition 103
system common definition 98
system service common information definition 149
transaction service definition 105
user service default definition 136
user service definition 143
XATMI communication service definition 122
max_socket_msg
user service default definition 136
user service definition 142
max_socket_msglen
user service default definition 136
user service definition 142
MCF communication service
partially starting (mcftstart) 228
partially stopping (mcftstop) 228
referencing status of (mcftlscom) 228
MCF statistics
editing (mcfreport) 228
outputting (jnlmcst) 221
outputting (mcfstats) 228
MCF trace file, forcibly swapping (mcftswptr) 228
MCF trace information
ending acquisition of (mcftstptr) 228
starting acquisition of (mcftstrtr) 228
mcf_jnl_buff_size
user service default definition 135
user service definition 141
mcf_mgrid
user service default definition 136
user service definition 142
mcf_prf_trace
definition of system service information 149
user service default definition 139
user service definition 146
mcf_prf_trace_level (system service common information definition) 149
mcf_psv_id
user service default definition 135
user service definition 142
mcf_service_max_count
user service default definition 136
user service definition 142
mcf_spp_oj
user service default definition 137
user service definition 143
mcfaactap (releases applications from shutdown status) 225
mcfaalcap (MCF application definition) 148

mcfaclcap (initializes the application abnormal termination count) 226
mcfadctap (shuts down applications) 225
mcfadltap (deletes application timer start requests) 226
mcfaenv (MCF application definition) 148
mcfalsap (displays application status) 225
mcfalstap (displays application timer start requests) 226
mcfmcname (MCF manager definition) 147
mcfmcomn (MCF manager definition) 147
mcfmenv (MCF manager definition) 147
mcfmexp (MCF manager definition) 147
mcfmqgid (MCF manager definition) 147
mcfmsmsg (MCF manager definition) 147
mcfmsts (MCF manager definition) 147
mcfmsvg (MCF manager definition) 147
mcfmuap (MCF manager definition) 147
mcfreport (edits MCF statistics) 228
mcfstats (outputs MCF statistics) 228
mcftactcn (establishes connections) 225
mcftactle (releases logical terminals from shutdown status) 226
mcftactmj (starts acquisition of logical terminal-related message journals) 226
mcftactsg (releases service groups from shutdown status) 227
mcftactss (starts a session) 227
mcftactsv (releases services from shutdown status) 227
mcftalcle (MCF communication configuration definition) 148
mcftbuf (MCF communication configuration definition) 148
mcftchcn (changes connections) 225
mcftcomn (MCF communication configuration definition) 147
mcftdctcn (releases connections) 225
mcftdctle (shuts down logical terminals) 226
mcftdctmj (ends acquisition of logical terminal-related message journals) 226
mcftdctsg (shuts down service groups) 227
mcftdctss (ends a session) 227
mcftdctsv (shuts down services) 227
mcftdlqle (deletes output queues for logical terminals) 226
mcftdlqsg (deletes input queues for service groups) 227
mcftdmpqu (copies the contents of the input/output queues) 228
mcftedalt (ends alternate sending) 227
mcftendct (forcibly ends continuous inquiry-response processing for a logical terminal) 227
mcftenv (MCF communication configuration definition) 147
mcfthldiq (places input queue processing for service groups in hold status) 227
mcfthldoq (places output queue processing for a logical terminal in hold status) 226
mcftlsbuf (displays the usage status of a buffer group) 227
mcftlscn (displays connection status) 225
mcftlscom (references the status of an MCF communication service) 228
mcftlsle (displays logical terminal status) 226
mcftlsln (displays network status) 225
mcftlssg (displays the status of service groups) 227
mcftlssv (displays service status) 227
mcftlstrd (displays concurrent message processing status) 225
mcftlsutm (displays the status of user timer monitoring) 228
mcftofln (ends acceptance of server-type connection establishment requests) 225
mcftonln (starts acceptance of server-type connection establishment requests) 225
mcftped (MCF communication configuration definition) 148
mcftpsvr (MCF communication configuration definition) 148
mcftrlsiq (releases input queue processing for service groups from hold status) 227
mcftrlsoq (releases output queue processing for a logical terminal from hold status) 226
mcftspqle (skips the first message in the message queue for a logical terminal) 226
mcftstalt (starts alternate sending) 227

mcftstart (partially starts an MCF communication service) 228
mcftstop (partially stops an MCF communication service) 228
mcftstptr (ends acquisition of MCF trace information) 228
mcftstrtr (starts acquisition of MCF trace information) 228
mcftsts (MCF communication configuration definition) 148
mcftswptr (forcibly swaps MCF trace files) 228
mcfttim (MCF communication configuration definition) 148
mcfttrc (MCF communication configuration definition) 148
mcfttred (MCF communication configuration definition) 147
mcfuevt (starts an application program) 226
message log file, displaying contents of (logcat) 218
message log, changing setting for real-time output facility for (logcon) 218
message queue
    allocating physical file for (queinit) 223
    deleting physical file for (querm) 223
message_buflen
    user service default definition 134
    user service definition 141
message_cell_size
    user service default definition 136
    user service definition 142
message_store_buflen
    client service definition 119
    user service default definition 135
    user service definition 141
mode_conf (system environment definition) 95
module
    definition of system service information 149
    user service definition 140
mqa_conf (system service configuration definition) 97
mrs_conf (system service configuration definition) 96
multi-node area or subarea
    starting (dcmstart) 213
    terminating (dcmstop) 213
multi_node_option (system common definition) 98
multi_schedule
    user service default definition 137
    user service definition 144
multiOpenTP1
    canceling setup of (dcsetupml) 228
    checking setup status of (dcdls) 228
    setting up (dcsetupml) 228
multiOpenTP1 console, creating (dcmakecon) 228
multiOpenTP1 operation 179
multiOpenTP1 setting 90
my_host (system common definition) 98

N

nam_prf_trace_level (system common definition) 100
namalivechk (checks OpenTP1 startup and clears cache) 217
namblad (manipulates the RPC suppression list) 218
namchgfl (changes the domain configuration (using domain definition files) 217
namdomainsetup (registers and deletes the domain representative schedule service) 217
name_audit_conf (name service definition) 102
name_audit_interval (name service definition) 102
name_audit_watch_time (name service definition) 102
name_cache_size (name service definition) 101
name_cache_validity_time (name service definition) 102
name_domain_file_use (system common definition) 99
name_global_lookup (name service definition) 102
name_nodeid_check_message (name service definition) 102
name_notify (system common definition) 99
name_port (system common definition) 97
name_rpc_control_list (name service definition) 102
name_service_extend (name service definition) 102
name_total_size (name service definition) 101
namndchg (changes the domain configuration (using the system common definition)) 217
namsvinf (displays OpenTP1 server information) 217

namunavl (forcibly disables start notification information) 217
network status, displaying (mcftlsln) 225
nice
    RAP-processing listener service definition 124
    user service default definition 134
    user service definition 141
njsmkdll (creates a transaction control DLL) 229
nl_watch_time (global archive journal service definition) 115
node_down_restart
    user service default definition 136
    user service definition 142
node_id (system common definition) 97
ntbstart (starts OpenTP1) 229
ntbtail (outputs the contents of a specified file to the console) 229

O

open_rm (client service definition) 119
OpenTP1
    deleting, from OS (dcsetup) 213
    registering, into OS (dcsetup) 213
    reserving and releasing resource for internal control (dcmakeup) 213
    starting (dcstart) 213
    starting (ntbstart) 229
    terminating (dcstop) 213
OpenTP1 administrator, registering 57
OpenTP1 command, executing from scenario template (dcjcmdex) 213
OpenTP1 console output facility 16
OpenTP1 Control window 207
OpenTP1 Environment Setup window (Output) 202
OpenTP1 Environment window 197
    parameter information 199
OpenTP1 file group, changing (filchgrp) 218
OpenTP1 file owner, changing (filchown) 218
OpenTP1 file system 62
    backup (filbkup) 218
    configuring 62
    configuring by applying raw I/O facility 65
    displaying content of (fills) 218
    displaying status of (filstatfs) 218
    initializing (filmkfs) 218
    restoring (filrstr) 218
OpenTP1 file, changing access permission mode for (filchmod) 218
OpenTP1 for Windows
    GUI for starting 205
    GUI for terminating 205
    GUI provided by 195
    note about environment that uses both Visual Studio and COBOL 45
    note about termination 178
    note about UAP execution in single-thread environment 44
    note about UAP termination 44
    note about using Net Express 45
    note about using Visual Studio 44
    note on programming 44
    note on stub creation 44
    overview of operation 172
    starting 173
    starting, from Windows Services dialog box 173
    starting, with command 174
    starting, with GUI 175
    terminating 177
    terminating, from Windows Services dialog box 177
    terminating, with command 177
    terminating, with GUI 178
OpenTP1 node, displaying status of (dcndls) 214
OpenTP1 process, forcibly terminating (prckill) 215
OpenTP1 processes, list of 266
OpenTP1 server information, displaying (namsvinf) 217
OpenTP1 startup, checking (namalivechk) 217
OpenTP1 system definition information, creating 58
OpenTP1_Deploy 187
OpenTP1_ScenarioScaleout 188
OpenTP1_Undeploy 187
operand specified in order to use function for checking standard output and standard error, relationships among 17
operation commands, list of 212

Option dialog box 208
OS setting 88

P

parallel_count
client service definition 118
user service default definition 134
user service definition 141
permission control facility 18
physical file
backup (dambkup) 222
deleting (damdel) 222
initializing (damload) 222
restoring (damrstr) 222
polling_control_data
transaction service definition 105
user service default definition 138
user service definition 144
pool information from lock control table, displaying (lckpool) 217
prc_abort_signal
user service default definition 136
user service definition 143
prc_corecompress (process service definition) 103
prc_coresave_path (process service definition) 102
prc_current_work_path (system common definition) 99
prc_port (system common definition) 98
prc_prf_trace (process service definition) 102
prc_process_count (process service definition) 102
prc_recovery_resident (process service definition) 102
prc_take_over_svpath (process service definition) 102
prckill (forcibly terminates one or more OpenTP1 processes) 215
prcls (displays the status of one or more servers) 215
prcpath (changes one or more search path names for the user server and for the commands started from the user server) 215
prcpathls (displays the search path names for the user server and for the commands started from the user server) 215
prcsvpath 158
process service definition 103
prctctrl (terminates and restarts the prctee process) 214
prctee (redirects standard output and standard error) 214
prctee process, terminating and restarting (prctctrl) 214
preend_warning_watch_time (system environment definition) 95
prf_file_count
definition of performance verification trace 127
JNL performance verification trace definition 128
LCK performance verification trace definition 128
MCF performance verification trace definition 148
TRN event trace definition 129
XAR performance verification trace definition 127
prf_file_size
definition of performance verification trace 127
JNL performance verification trace definition 128
LCK performance verification trace definition 128
MCF performance verification trace definition 148
TRN event trace definition 128
XAR performance verification trace definition 127
prf_information_level
definition of performance verification trace 127
LCK performance verification trace definition 128
TRN event trace definition 129
XAR performance verification trace definition 127
prf_trace (system common definition) 98
prf_trace_backup

definition of performance verification trace 127
JNL performance verification trace definition 128
prfed (edits and outputs trace information files) 224
prfget (gets trace information files) 224
process
restarting (scdrsprc) 216
terminating (scdrsprc) 216
process service definition 158
process service, restarting and updating from definition (dcreset) 213
process_privilege_name
user service default definition 139
user service definition 146
process_privilege_restrict
user service default definition 139
user service definition 146
processes, changing number of (scdchprc) 216
product information, displaying (dcpplist) 215
purge_msgget
user service default definition 136
user service definition 142

Q

que_conf (system service configuration definition) 96
que_io_maxrecsize (MCF-message queue service definition) 122
que_xidnum (MCF-message queue service definition) 122
quegrp (MCF-message queue service definition) 123
queinit (allocates a physical file for the message queue) 223
quels (displays the status of queue groups) 223
querm (deletes a physical file for the message queue) 223
queue group, displaying status of (quels) 223

R

RAP-processing listener and server, displaying status of (rapls) 224
RAP-processing listener service definition 161
rap_autoconnect_con_error_msg
user service default definition 138
user service definition 145
rap_client_manager_node (RAP-processing listener service definition) 125
rap_client_manager_port (RAP-processing client manager service definition) 126
rap_connect_interval (RAP-processing listener service definition) 125
rap_connect_retry_count (RAP-processing listener service definition) 125
rap_connect_retry_interval (RAP-processing listener service definition) 125
rap_connection_assign_type (RAP-processing listener service definition) 125
rap_inquire_time (RAP-processing listener service definition) 124
rap_inquire_timeout_message (RAP-processing listener service definition) 125
rap_io_retry_interval (RAP-processing listener service definition) 125
rap_listen_backlog (RAP-processing listener service definition) 125
rap_listen_inf (RAP-processing client manager service definition) 126
rap_listen_port (RAP-processing listener service definition) 123
rap_max_buff_size (RAP-processing listener service definition) 125
rap_max_client (RAP-processing listener service definition) 125
rap_message_id_change_level
RAP-processing listener service definition 126
user service default definition 139
user service definition 146
rap_msg_output_interval (RAP-processing listener service definition) 125
rap_notify (RAP-processing listener service definition) 125
rap_parallel_server (RAP-processing listener service definition) 124
rap_recovery_server (RAP-processing listener service definition) 125

rap_sock_count (RAP-processing listener service definition) 125
rap_sock_interval (RAP-processing listener service definition) 125
rap_stay_warning_interval (RAP-processing listener service definition) 126
rap_stay_watch_time (RAP-processing listener service definition) 126
rap_term_disconnect_time (RAP-processing listener service definition) 126
rap_watch_time
    RAP-processing client manager service definition 126
    RAP-processing listener service definition 124
rapdfgen (automatically generates the definitions used for the remote API facility) 224
rapls (displays the status of a RAP-processing listener and server) 224
rapsetup (sets up an execution environment for the remote API facility) 224
raw I/O facility 65
    configuring OpenTP1 file system by applying 65
real-time statistics
    changing setting for (rtsstats) 225
    outputting, to standard output (rtsls) 225
real-time statistics service, setting up execution environment for (rtssetup) 225
receive_from
    user service default definition 135
    user service definition 141
redirect_file (system environment definition) 95
redirect_file_name (system environment definition) 95
redirect_file_size (system environment definition) 95
remote API facility
    automatically generating definition used for (rapdfgen) 224
    setting up execution environment for (rapsetup) 224
resource group connection, forcibly releasing (jnlardis) 222
resource manager information, displaying (trnlsrm) 223
resource manager, registering (trnlnkrm) 223
rmm_abort_command (monitored RM definition) 121
rmm_check_services (RMM service definition) 120
rmm_command_gid (monitored RM definition) 121
rmm_command_uid (monitored RM definition) 121
rmm_command_watch_time (monitored RM definition) 121
rmm_conf (system service configuration definition) 96
rmm_down_with_system (RMM service definition) 120
rmm_get_pid_command (monitored RM definition) 121
rmm_start_command (monitored RM definition) 120
rmm_start_watch_time (monitored RM definition) 121
rmm_stop_command (monitored RM definition) 120
rmm_sysdown_with_rm (RMM service definition) 120
rmm_system_behavior (RMM service definition) 120
RPC suppression list, manipulating (namblad) 218
RPC trace information
    merging (rpcmrg) 224
    outputting (rpcdump) 224
rpc_buffer_pool_max
    user service default definition 138
    user service definition 145
rpc_datacomp (system common definition) 98
rpc_delay_statistics (system common definition) 98
rpc_destination_mode
    user service default definition 137
    user service definition 144
rpc_extend_function
    RAP-processing listener service definition 125
    user service default definition 136
    user service definition 143
rpc_max_message_size (system common definition) 99
rpc_message_level (system common definition) 98

rpc_multi_tp1_in_same_host (system common definition) 98
rpc_netmask (system common definition) 98
rpc_port_base (system common definition) 98
rpc_rap_auto_connect
    user service default definition 137
    user service definition 144
rpc_rap_inquire_time
    user service default definition 137
    user service definition 144
rpc_request_cancel_for_timedout
    user service default definition 137
    user service definition 144
rpc_response_statistics
    RAP-processing listener service definition 124
    user service default definition 136
    user service definition 142
rpc_retry (system common definition) 97
rpc_retry_count (system common definition) 98
rpc_retry_interval (system common definition) 98
rpc_router_retry_count (system common definition) 99
rpc_router_retry_interval (system common definition) 99
rpc_send_retry_count
    system common definition 99
    user service default definition 138
    user service definition 144
rpc_send_retry_interval
    system common definition 99
    user service default definition 138
    user service definition 144
rpc_server_busy_count (system common definition) 99
rpc_service_retry_count
    user service default definition 136
    user service definition 143
rpc_trace
    RAP-processing listener service definition 124
    system common definition 97
    user service default definition 137
    user service definition 143
rpc_trace_name
    RAP-processing listener service definition 124
    system common definition 97
    user service default definition 137
    user service definition 143
rpc_trace_size
    RAP-processing listener service definition 124
    system common definition 97
    user service default definition 137
    user service definition 143
rpcdump (outputs RPC trace information) 224
rpcmrg (merges RPC trace information) 224
rpcstat (displays the status of temporary close processing) 214
RTS log file, editing and outputting (rtsedit) 225
rts_cpd_collct_cpd (real-time acquisition item definition) 130
rts_cpd_validt_cpd (real-time acquisition item definition) 130
rts_dam_cache_block (real-time acquisition item definition) 133
rts_dam_fj (real-time acquisition item definition) 132
rts_dam_read (real-time acquisition item definition) 132
rts_dam_read_err (real-time acquisition item definition) 132
rts_dam_shm_pool (real-time acquisition item definition) 133
rts_dam_trn_branch (real-time acquisition item definition) 132
rts_dam_write (real-time acquisition item definition) 132
rts_dam_write_err (real-time acquisition item definition) 132
rts_item_max (real-time statistics service definition) 129
rts_jnl_buf_full (real-time acquisition item definition) 130
rts_jnl_io_wait (real-time acquisition item definition) 130
rts_jnl_jnl_input (real-time acquisition item definition) 130

rts_jnl_jnl_output (real-time acquisition item definition) 130
rts_jnl_read (real-time acquisition item definition) 130
rts_jnl_swap (real-time acquisition item definition) 130
rts_jnl_wait_buf (real-time acquisition item definition) 130
rts_jnl_write (real-time acquisition item definition) 130
rts_lck_deadlock (real-time acquisition item definition) 131
rts_lck_lock_acqst (real-time acquisition item definition) 130
rts_lck_lock_wait (real-time acquisition item definition) 130
rts_log_file (real-time statistics service definition) 129
rts_log_file_backup (real-time statistics service definition) 129
rts_log_file_count (real-time statistics service definition) 129
rts_log_file_name (real-time statistics service definition) 129
rts_log_file_size (real-time statistics service definition) 129
rts_mcf_ap_scd_stay (real-time acquisition item definition) 134
rts_mcf_ap_usr_srvc (real-time acquisition item definition) 134
rts_mcf_in_msg_scd_wait (real-time acquisition item definition) 134
rts_mcf_out_msg_norm_scd_wait (real-time acquisition item definition) 134
rts_mcf_out_msg_prio_scd_wait (real-time acquisition item definition) 134
rts_mcf_out_msg_resp_scd_wait (real-time acquisition item definition) 134
rts_mcf_out_msg_sync_scd_wait (real-time acquisition item definition) 134
rts_mcf_que_scd_wait_num (real-time acquisition item definition) 134
rts_nam_global_cache_hit (real-time acquisition item definition) 131
rts_nam_local_cache_hit (real-time acquisition item definition) 131
rts_nam_lookup (real-time acquisition item definition) 131
rts_nam_node_lookup (real-time acquisition item definition) 131
rts_nam_node_lookup_responce (real-time acquisition item definition) 131
rts_osl_dynmem_acq (real-time acquisition item definition) 131
rts_osl_dynmem_pol (real-time acquisition item definition) 131
rts_osl_stamem_acq (real-time acquisition item definition) 131
rts_osl_stamem_pol (real-time acquisition item definition) 131
rts_prc_prc_genert (real-time acquisition item definition) 131
rts_prc_prc_num (real-time acquisition item definition) 131
rts_prc_prc_term (real-time acquisition item definition) 131
rts_prc_sys_abnml (real-time acquisition item definition) 131
rts_prc_uap_abnml (real-time acquisition item definition) 131
rts_que_delay_msg (real-time acquisition item definition) 131
rts_que_delay_rec (real-time acquisition item definition) 131
rts_que_delay_wrt (real-time acquisition item definition) 131
rts_que_read (real-time acquisition item definition) 131
rts_que_read_err (real-time acquisition item definition) 131
rts_que_real_read (real-time acquisition item definition) 131
rts_que_real_write (real-time acquisition item definition) 131
rts_que_wait_buf (real-time acquisition item definition) 131
rts_que_write (real-time acquisition item definition) 131

rts_que_write_err (real-time acquisition item definition) 131
rts_rpc_rpc_call (real-time acquisition item definition) 132
rts_rpc_rpc_call_chained (real-time acquisition item definition) 132
rts_rpc_rpc_ovrtim (real-time acquisition item definition) 132
rts_rpc_usr_srvc (real-time acquisition item definition) 132
rts_scd_lack_buf (real-time acquisition item definition) 132
rts_scd_parallel (real-time acquisition item definition) 132
rts_scd_scd_stay (real-time acquisition item definition) 132
rts_scd_scd_wait (real-time acquisition item definition) 132
rts_scd_schedule (real-time acquisition item definition) 132
rts_scd_svc_scd_wait (real-time acquisition item definition) 132
rts_scd_svc_using_buf (real-time acquisition item definition) 132
rts_scd_using_buf (real-time acquisition item definition) 132
rts_service_max (real-time statistics service definition) 129
rts_swap_message (real-time statistics service definition) 129
rts_tam_read (real-time acquisition item definition) 133
rts_tam_read_err (real-time acquisition item definition) 133
rts_tam_real_renew (real-time acquisition item definition) 133
rts_tam_real_renew_time (real-time acquisition item definition) 133
rts_tam_rec_refer (real-time acquisition item definition) 133
rts_tam_rec_renew (real-time acquisition item definition) 133
rts_tam_write (real-time acquisition item definition) 133
rts_tam_write_err (real-time acquisition item definition) 133
rts_trcput_interval (real-time statistics service definition) 129
rts_trn_branch (real-time acquisition item definition) 132
rts_trn_cmt_cmd (real-time acquisition item definition) 132
rts_trn_commit (real-time acquisition item definition) 132
rts_trn_haz_cmd (real-time acquisition item definition) 132
rts_trn_mix_cmd (real-time acquisition item definition) 132
rts_trn_rbk_cmd (real-time acquisition item definition) 132
rts_trn_rollback (real-time acquisition item definition) 132
rts_trn_sync_point (real-time acquisition item definition) 132
rts_xar_call (real-time acquisition item definition) 133
rts_xar_call_err (real-time acquisition item definition) 133
rts_xar_commit (real-time acquisition item definition) 133
rts_xar_commit_err (real-time acquisition item definition) 133
rts_xar_end (real-time acquisition item definition) 133
rts_xar_end_err (real-time acquisition item definition) 133
rts_xar_forget (real-time acquisition item definition) 133
rts_xar_forget_err (real-time acquisition item definition) 133
rts_xar_prepare (real-time acquisition item definition) 133
rts_xar_prepare_err (real-time acquisition item definition) 133
rts_xar_recover (real-time acquisition item definition) 133
rts_xar_recover_err (real-time acquisition item definition) 133

rts_xar_rollback (real-time acquisition item definition) 133
rts_xar_rollback_err (real-time acquisition item definition) 133
rts_xar_start (real-time acquisition item definition) 133
rts_xar_start_err (real-time acquisition item definition) 133
rtsedit (edits and outputs RTS log files) 225
rtsls (outputs real-time statistics to standard output) 225
rtsput (real-time statistics service definition) 130
rtssetup (sets up an execution environment for the real-time statistics service) 225
rtsstats (changes settings for real-time statistics) 225
runtime library path, setting 58

S

sample program 46
sample scenario template
    name and processing of 183
    registering 185
    using 182
scale-out operation 180
scd_advertise_control (schedule service definition) 103
scd_announce_server_status (schedule service definition) 103
scd_hold_recovery (schedule service definition) 103
scd_hold_recovery_count (schedule service definition) 103
scd_message_level (schedule service definition) 103
scd_pool_warning_interval
    user service default definition 139
    user service definition 145
scd_pool_warning_use_rate
    user service default definition 139
    user service definition 145
scd_poolfull_check_count
    user service default definition 139
    user service definition 145
scd_poolfull_check_interval
    user service default definition 139
    user service definition 145
scd_port (schedule service definition) 103
scd_retry_of_comm_error (schedule service definition) 103
scd_server_count (schedule service definition) 103
scd_this_node_first (schedule service definition) 103
scdbufgrp
    schedule service definition 104
    user service default definition 140
    user service definition 146
scdchprc (changes the number of processes) 216
scdhold (shuts down scheduling) 215
scdls (displays scheduling status) 215
scdmulti
    schedule service definition 104
    user service default definition 140
    user service definition 146
scdrles (restarts scheduling) 216
scdrsprc (terminates and restarts processes) 216
scdsvcdef
    user service default definition 140
    user service definition 146
scenario template
    details of 186
    system operation using 180
scenario template definition file 180
scenario, registering 180
schedule_delay_abort
    user service default definition 138
    user service definition 145
schedule_delay_limit
    user service default definition 138
    user service definition 145
schedule_method
    user service default definition 137
    user service definition 143
schedule_priority
    user service default definition 134
    user service definition 141
schedule_rate (schedule service definition) 103
scheduling
    displaying status of (scdls) 215
    restarting (scdrles) 216
    shutting down (scdhold) 215

search path name for user server and for command started from user server
changing (prcpath) 215
displaying (prcpathls) 215
server
displaying status of (prcls) 215
starting (dcsvstart) 215
terminating (dcsvstop) 215
server-type connection establishment request
ending acceptance of (mcftofln) 225
starting acceptance of (mcftonln) 225
server_count (system environment definition) 95
server_type
user service default definition 136
user service definition 142
service
displaying status of (mcftlssv) 227
releasing, from shutdown status (mcftactsv) 227
shutting down (mcftdctsv) 227
user service definition 140
service group
deleting input queue for (mcftdlqsg) 227
displaying status of (mcftlssg) 227
from hold status, releasing input queue processing for (mcftrlsiq) 227
in hold status, placing input queue processing for (mcfthldiq) 227
releasing, from shutdown status (mcftactsg) 227
shutting down (mcftdctsg) 227
service_expiration_time
user service default definition 137
user service definition 144
service_group (user service definition) 140
service_hold
user service default definition 135
user service definition 142
service_priority_control
user service default definition 135
user service definition 142
service_term_watch_time
user service default definition 136
user service definition 143
service_wait_time
user service default definition 137
user service definition 143
session
ending (mcftdctss) 227
starting (mcftactss) 227
shared memory dump, outputting (usmdump) 224
shared memory, displaying status of (dcshmls) 214
shmpool_attribute (system environment definition) 95
specified file to console, outputting contents of (ntbtail) 229
standard error
functions for checking 13
redirecting (prctee) 214
relationships among operands specified in order to use functions for checking 17
standard output
functions for checking 13
redirecting (prctee) 214
relationships among operands specified in order to use functions for checking 17
standard output redirect facility 13
standard output redirect file to console, outputting (tp1console) 229
start notification information, forcibly disabling (namunavl) 217
start_scheduling_timing (system environment definition) 95
startup mode, determining 175
static_shmpool_size (system environment definition) 95
statistical information, outputting (jnlstts) 221
statistics (system common definition) 99
status file
closing (stsclose) 219
creating (stsinit) 218
deleting (stsrm) 219
displaying content of (stsfills) 219
displaying status of (stsls) 219
initializing (stsinit) 218
opening (stsopen) 219
swapping (stsswap) 219
status_change_when_terming

user service default definition 137
user service definition 144
stay_watch_abort
user service default definition 139
user service definition 145
stay_watch_check_interval
user service default definition 139
user service definition 145
stay_watch_check_rate
user service default definition 139
user service definition 145
stay_watch_queue_count
user service default definition 139
user service definition 145
stay_watch_start_interval
user service default definition 139
user service definition 145
sts_file_name_1 (status service definition) 108
sts_file_name_2 (status service definition) 108
sts_file_name_3 (status service definition) 108
sts_file_name_4 (status service definition) 108
sts_file_name_5 (status service definition) 108
sts_file_name_6 (status service definition) 108
sts_file_name_7 (status service definition) 108
sts_initial_error_switch (status service definition) 108
sts_last_active_file (status service definition) 108
sts_last_active_side (status service definition) 108
sts_single_operation_switch (status service definition) 108
stsclose (closes a status file) 219
stsfills (displays the contents of a status file) 219
stsinit (creates and initializes a status file) 218
stsls (displays the status of status files) 219
stsopen (opens a status file) 219
stsrm (deletes a status file) 219
stsswap (swaps status files) 219
system common definition 156
system definition
changing 58
checking (dcdefchk) 214
format of explanation 150
system definition operand, specifying (dcjchconf) 213
system definitions, list of 94
system environment definition 152
system operation, using scenario template 180
system service information, definition of 170
system statistical information
editing and outputting, to standard output in real time (dcreport) 214
starting acquisition of (dcstats) 213
terminating acquisition of (dcstats) 213
system_id (system common definition) 97
system_init_watch_time (system environment definition) 95
system_terminate_watch_time (system environment definition) 95

T

TAM file
backup (tambkup) 223
deleting (tamdel) 223
displaying synonym information for hash-format (tamhsls) 223
initializing (tamcre) 222
recovering (tamfrc) 223
restoring (tamrstr) 223
TAM locked resource name, converting (tamlckls) 223
TAM table
adding (tamadd) 222
detaching (tamrm) 222
displaying status of (tamls) 222
displaying synonym information for hash-format (tamhsls) 223
loading (tamload) 223
logically shutting down (tamhold) 222
releasing from shutdown status (tamrles) 222
unloading (tamunload) 223
tam_cbl_level (TAM service definition) 118
tam_conf (system service configuration definition) 96
tam_jnl_err_flag (TAM service definition) 118
tam_max_filesize (TAM service definition) 118
tam_max_recsize (TAM service definition) 118
tam_max_tblnum (TAM service definition) 117
tam_max_trnfilnum (TAM service definition) 118

tam_max_trnnum (TAM service definition) 118
tam_pool_attri (TAM service definition) 118
tam_tbl_lock_mode (TAM service definition) 118
tamadd (adds a TAM table) 222
tambkup (backs up a TAM file) 223
tamcre (initializes a TAM file) 222
tamdel (deletes a TAM file) 223
tamfrc (recovers a TAM file) 223
tamhold (logically shuts down a TAM table) 222
tamhsls (displays synonym information for a hash-format TAM file and table) 223
tamlckls (converts TAM locked resource names) 223
tamload (loads a TAM table) 223
tamls (displays the status of TAM tables) 222
tamrles (releases a TAM table from shutdown status) 222
tamrm (detaches a TAM table) 222
tamrstr (restores a TAM file) 223
tamtable (TAM service definition) 118
tamunload (unloads a TAM table) 223
temporary close processing, displaying status of (rpcstat) 214
term_watch_count (process service definition) 102
term_watch_time
    process service definition 102
    user service default definition 135
    user service definition 141
termed_after_service
    user service default definition 136
    user service definition 143
thdlock_sleep_time
    system common definition 99
    system service common information definition 149
thread_stack_size (transaction service definition) 105
thread_yield_interval
    transaction service definition 105
    user service default definition 138
    user service definition 145
tim_watch_count (timer service definition) 101
timeout information file, deleting (lckrminf) 217
tp1console (outputs the standard output redirect file to the console) 229
tptrnls (displays undetermined-transaction information for OSI TP communication) 216
trace information file
    editing and outputting (prfed) 224
    getting (prfget) 224
transaction
    committing (trncmt) 216
    displaying status of (trnls) 216
    forcibly terminating (trnfgt) 216
    rollback (trnrbk) 216
transaction control DLL, creating (njsmkdll) 229
transaction control object file, creating (trnmkobj) 223
transaction statistical information
    starting acquisition of (trnstics) 216
    terminating acquisition of (trnstics) 216
trf_put
    RAP-processing listener service definition 124
    user service default definition 136
    user service definition 142
trn_completion_limit_time
    client service definition 119
    RAP-processing listener service definition 125
    transaction service definition 106
    user service default definition 139
    user service definition 145
trn_cpu_time
    client service definition 119
    RAP-processing listener service definition 124
    transaction service definition 104
    user service default definition 135
    user service definition 142
trn_crm_use (transaction service definition) 105
trn_expiration_time
    client service definition 118
    RAP-processing listener service definition 124
    transaction service definition 104
    user service default definition 135
    user service definition 141
trn_expiration_time_suspend

client service definition 119
RAP-processing listener service definition 124
transaction service definition 104
user service default definition 135
user service definition 141
trn_extend_function (transaction service definition) 106
trn_limit_time
client service definition 119
RAP-processing listener service definition 124
transaction service definition 105
user service default definition 137
user service definition 143
trn_max_crm_subordinate_count (transaction service definition) 105
trn_max_subordinate_count (transaction service definition) 104
trn_optimum_item
client service definition 119
RAP-processing listener service definition 124
transaction service definition 105
user service default definition 136
user service definition 142
trn_partial_recovery_type
client service definition 119
RAP-processing listener service definition 125
transaction service definition 105
user service default definition 137
user service definition 143
trn_prf_event_trace_condition (transaction service definition) 106
trn_prf_event_trace_level (transaction service definition) 106
trn_prf_trace_level (system common definition) 98
trn_processing_in_rm_error (transaction service definition) 105
trn_recovery_failmsg_interval (transaction service definition) 105
trn_recovery_list_remove (transaction service definition) 105
trn_recovery_list_remove_level (transaction service definition) 105
trn_recovery_process_count (transaction service definition) 104
trn_retry_count_rm_open (transaction service definition) 105
trn_retry_interval_rm_open (transaction service definition) 105
trn_rm_open_close_scope
transaction service definition 105
user service default definition 136
user service definition 142
trn_rollback_information_put
client service definition 119
RAP-processing listener service definition 124
transaction service definition 105
user service default definition 137
user service definition 143
trn_rollback_response_receive
client service definition 119
RAP-processing listener service definition 125
transaction service definition 105
user service default definition 137
user service definition 143
trn_start_recovery_interval (transaction service definition) 106
trn_start_recovery_mode (transaction service definition) 106
trn_start_recovery_watch_time (transaction service definition) 106
trn_statistics_item
client service definition 119
RAP-processing listener service definition 124
transaction service definition 104
user service default definition 136
user service definition 142
trn_tran_process_count (transaction service definition) 104
trn_tran_recovery_list (transaction service definition) 104
trn_tran_statistics (transaction service definition) 104

trn_wait_rm_open (transaction service definition) 105
trn_watch_time
client service definition 119
RAP-processing listener service definition 124
transaction service definition 105
user service default definition 137
user service definition 143
trn_xa_commit_error (transaction service definition) 106
trn_xar_use (transaction service definition) 106
trncmt (commits transactions) 216
trndlinf (deletes undetermined-transaction information files) 216
trnfgt (forcibly terminates transactions) 216
trnlnkrm
extended RM registration definition 121
registers resource managers 223
trnls (displays transaction status) 216
trnlsrm (displays resource manager information) 223
trnmkobj (creates a transaction control object file) 223
trnrbk (rolls back transactions) 216
trnrmid
user service default definition 139
user service definition 146
trnstics (starts and terminates acquisition of transaction statistical information) 216
trnstring (transaction service definition) 106
troubleshooting 190
action in event of application exception 191
action in event of shortage of ports (tuning number of TCP/IP ports) 192
response when user server shuts down immediately after startup 59
using information acquired in event of failure 193
troubleshooting information, deleting (dccspool) 214
type
user service default definition 135
user service definition 141
TZ (log service definition) 114

U

UAP
compiling and linking 33
creating by using Net Express 40
creating for Visual Studio projects 35
creating in C language 33
creating in COBOL2002 36
UAP shared library, when using dynamic loading of service functions 48
UAP trace information, editing and outputting (uatdump) 224
uap_conf (system service configuration definition) 96
uap_trace_file_put
system common definition 100
user service default definition 135
user service definition 141
uap_trace_max
RAP-processing listener service definition 124
user service default definition 135
user service definition 141
uatdump (edits and outputs UAP trace information) 224
uid
RAP-processing client manager service definition 126
RAP-processing listener service definition 124
user service default definition 135
user service definition 142
undetermined-transaction information file, deleting (trndlinf) 216
undetermined-transaction information for OSI TP communication, displaying (tptrnls) 216
unload journal file
copying (jnlcopy) 220
editing and outputting (jnledit) 221
outputting record from (jnlrput) 221
sorting and merging, in time order (jnlsort) 221
user service default definition 163
user service definition 166
user service process, replacing service functions of 50

user timer monitoring, displaying status of (mcftlsutm) 228
user_command (system environment definition) 95
user_command_online (system environment definition) 95
user_command_online_tp1mngr_id (system environment definition) 95
user_server_ha (system environment definition) 95
usmdump (outputs a shared memory dump) 224

V

Visual Studio project settings 36

W

watch_next_chain_time
    user service default definition 135
    user service definition 141
watch_time
    checkpoint dump service definition 111
    client service definition 119
    DAM service definition 117
    interval service definition 107
    journal service definition 109
    log service definition 113
    multi-node structure definition 114
    name service definition 102
    process service definition 103
    RAP-processing listener service definition 126
    schedule service definition 103
    status service definition 108
    system common definition 100
    system journal service definition 110
    TAM service definition 118
    transaction service definition 106
    user service default definition 139
    user service definition 146

X

XA resource service
    releasing, from shutdown status (xarrles) 217
    shutting down (xarhold) 216
XAR event trace information, displaying (xarevtr) 216
XAR file
    creating (xarinit) 216
    deleting (xarrm) 217
    displaying status of (xarfills) 216
XAR transaction information, displaying (xarls) 217
XAR transaction status, changing (xarforce) 216
xar_eventtrace_level (XA resource service definition) 107
xar_eventtrace_record (XA resource service definition) 107
xar_msdtc_use (XA resource service definition) 107
xar_prf_trace_level (XA resource service definition) 107
xar_session_time (XA resource service definition) 107
xarevtr (displays XAR event trace information) 216
xarfile (XA resource service definition) 107
xarfills (displays the status of a XAR file) 216
xarforce (changes the status of a XAR transaction) 216
xarhold (shuts down the XA resource service) 216
xarinit (creates a XAR file) 216
xarls (displays XAR transaction information) 217
xarrles (releases the XA resource service from shutdown status) 217
xarrm (deletes a XAR file) 217
xat_aso_con_event_svcname (XATMI communication service definition) 122
xat_aso_discon_event_svcname (XATMI communication service definition) 122
xat_aso_failure_event_svcname (XATMI communication service definition) 122
xat_conf (system service configuration definition) 96
XAT_CONNECT_RESP_TIME
    user service default definition 140
    user service definition 146
xat_connect_resp_time
    user service default definition 139
    user service definition 145
xat_osi_usr
    user service default definition 137
    user service definition 143

xat_trn_expiration_time
user service default definition 136
user service definition 143
xatinitaeq (XATMI communication service definition) 122
xatinitapt (XATMI communication service definition) 122
xatsrvadd (XATMI communication service definition) 122