VISUAL message center

debuggingAgent

Debugging Agent

Quick Reference

5.0

VDW

tango04 Computing Group

Solutions for Advancing People

VISUAL Message Center Debugging Agent Quick Reference

Copyright Notice

Document date: February 2011

Document version: 1.01

Product version: 5.0

Trademarks

Technical Support

For technical support visit our web site at www.tango04.com.

Tango/04 Computing Group S.L.
Avda. Meridiana 358, 5 A-B
Barcelona, 08027
Spain

Tel: +34 93 274 0051

Table of Contents

Chapter 4

Chapter 5

Chapter 6

Appendices

How to Use this Guide

This chapter explains how to use Tango/04 User Guides and understand the typographical conventions used in all Tango/04 documentation.

Typographical Conventions

The following conventional terms, text formats, and symbols are used throughout Tango/04 printed documentation:

Convention	Description
Boldface	Commands, on-screen buttons and menu options.
Blue Italic	References and links to other sections in the manual or further documentation containing relevant information.
Italic	Text displayed on screen, or variables where the user must substitute their own details.
`Monospace`	Input commands such as System i commands or code, or text that users must type in.
UPPERCASE	Keyboard keys, such as CTRL for the Control key and F5 for the function key that is labeled F5.
	**Notes** and useful additional information.
	**Tips** and hints that will improve the users experience of working with this product.
	**Important** additional information that the user is strongly advised to note.
	**Warning** information. Failure to take note of this information could potentially lead to serious problems.

Chapter 1
Shortcuts by Action Type

1.1 General Actions

Action	Shortcut
Refresh	F5
Run to Cursor	F4
Zoom In	Ctrl+A
Zoom Out	Ctrl+Z
Go Back	Ctrl+B (after hyperlink jump)
Go Next (Forward)	Ctrl+N (after going back from hyperlink jump)
Find	Ctrl+F
Find Next	Shift+F4 (F16)
Copy	Ctrl+C
Paste	Ctrl+V
Print	Ctrl+P
Go to top of source	Ctrl+Home
Go to end of source	Ctrl+End
Go to beginning of source line	Home
Go to end of source line	End
By-word move	Ctrl+left/right arrow keys
By character move	Left/right arrow keys
By line move	Up/Down arrow keys
By Page move	PgDown/PgUp keys
Information	F11
System Command	Shift+F9 (F21)
User Options	Shift+F1 (F13)

Action	Shortcut
Open Job	Ctrl+J
Open Object	Ctrl+O
Open Source Member	Ctrl+S
Open Chat Session	Ctrl+H
Open Source Entry Utility (SEU) or PC based Editor Utility	Shift+F2 (F14)
Add Program to Debug session	Shift+F3 (F15)
Prompt for Command Parameters	F4 on Command Line

1.1.1 General Actions when the cursor is located over an hyperlink

Action	Shortcut
Go to (Add) module/program to Debug session	Left Mouse Click (on module names)
Go to Procedure/Subroutine Definition	Letf Mouse Click (on procedure names)

1.2 Debug Actions

1.2.1 Debug actions when a variable name is highlighted on the source code Window

Action	Shortcut
Show / Change Variables	F2
Show Tooltip Variable	Right Mouse Click

1.2.2 Debug actions when the cursor is located at a breakable statement

Action	Shortcut
Set/Clear Breakpoint	F6
Set/Clear Conditional Breakpoint	Shift+F6 (F18)
Set/Clear Temporal Breakpoint	Ctrl+F6 (F18)
Contextual Menu	Right Mouse Click

1.2.3 General Debug actions

Action	Shortcut
Pause / Stop Execution	F3
Run to Cursor	F4
Switch Opened Module	F7
Step Into	F8

Action	Shortcut
Step Over	F9
Cursor at Command / Source.	F10
Run / Resume Execution	F12
Go to Next Breakpoint	Shift+F8 (F20)
Add All Breakpoints	Shift+F10 (F22)
Remove ALL Breakpoints	Shift+F11 (F23)
Go to Execution Point	Ctrl+G

1.2.4 Debug Actions on the Line Status Area

Action	Shortcut
Toggle Breakpoint On/Off	Left Mouse Click (on blue dot or breakpoint indication)
Set/Clear Conditional Breakpoint	Shift+Left Mouse Click (on blue dot or breakpoint indication)
Set/Clear Temporal Breakpoint	Ctrl+Left Mouse Click (on blue dot or breakpoint indication)
Show Condition	Right Mouse Click

1.3 Program Navigator Actions

Action	Shortcut
Go to (Add) module/program	Left Mouse Click (on module names)
Go to Procedure/SubRt/Label Definition	Left Mouse Click (on label/name)
Go to Reference	Left Mouse Click (on any references)
Contextual Menu	Right Mouse Click
Expand / Collapse	Double Click

Chapter 2
Function Key Assignments (by Function Key)

Shortcut	Action
F1	* (Reserved for Help)
F2	Display / Change Highlighted Variable in Window
F3	Pause / Stop Running Program
F4	Run to Cursor on Source Member. Prompt for Command Parameters on Command Line.
F5	Refresh
F6	Toggle Breakpoint
F7	Switch to Next Opened Module
F8	Step Into
F9	Step Over on Source Member Retrieve previous Command on Command Line.
F10	Toggle Cursor between Source and Command Line
F11	Object/Module Information
F12	Run / Resume Execution
F13	User Options
F14	Open Source Entry Utility (SEU) or PC based Editor Utility
F15	Add Program to Debug Project
F16	Find Next
F17	*
F18	Set/Clear Conditional Breakpoint
F19	Not Assigned

Shortcut	Action
F20	Go to Next Breakpoint
F21	System Command
F22	Add ALL Breakpoints
F23	Remove ALL Breakpoints
F24	*

(* = Not assigned on this Release)

Please note that the Function Key assignments are compatible with SEU and/or VISUAL Debugger 5250 Edition when possible.

Chapter 3
VISUAL Message Center Debugging Agent Commands

Commands are an alternative way to obtain results. In most cases, you will *not* want to use the commands for the point-and-click, icon based counterparts are much easier to use. However, commands are very powerful and in some cases there is no way to avoid using them (for instance, if you want to execute a system command in a debugged job: you will have to type it in).

To execute a Debugging Agent command, enter it in the Command Line Window over the source code (if it is not displayed, click on the **Unroll** button to the left of the Command Line tag).

Note
Commands in square brackets `[]` are optional

Command	Action
A / AV / ADDVAR <expression> [*HEX]	Adds the expression to the `WatchVar` List. You can do this using the Watch icon on the toolbar when an expression is highlighted, too.
AL / ADDLIST	Shows the `WatchVar` List if hidden
ATTR <variable list>	Shows attributes of one or more variables
B / BOT / BOTTOM	Go top of source (similar to Ctrl-Home)
BREAK / AT <statement list>	Set an unconditional breakpoint on one or more statements (use not recommended as point-and-click facilities are much easier to use)
BREAK / AT <statement list> WHEN <condition>	Set a conditional breakpoint on one or more statements (use not recommended as point-and-click features are much easier to use). Condition must follow the debugged language syntax conventions.
C / CV / CHGVAR <variable>	Change variable (shows window to enter new value) (similar to F2)
CL / CLRLIST	Removes all the expressions on the `WatchVar` List
CLEAR <statement list>	Removes one or multiple breakpoints
CLEAR PGM	Removes all breakpoints in the program

Command	Action
CLEAR WATCH <watch number list>	Removes one or more watches by number
CLEAR WATCH ALL	Removes all watches in the program
D / DV / DSPVAR <variable> [*HEX]	Display variable in Command Log window (similar to `EVAL [:X]`)
EVAL/LIST <variable or expression> [:u | :c: | :a | :x | :f | :s] [<lenght>]	Shows variable or expression value in Command Log window. For more information on formatting and expressions, see below.
F/FIND <string> [N/NEXT | P/PREV | F/FIRST | L/LAST | A/ALL]	`FIND` String
F/FIND <*BKP | *BKPNOEXE | *CHG | *COM | *NOCOM> [N/NEXT | P/PREV | F/FIRST | L/LAST | A/ALL]	`FIND` Extended
I / INFO	Program / Module Information
QUAL <statement>	Allows you to define the scope of variables that appear in subsequent `EVAL` or `WATCH` commands. Mostly needed when using local variables with the same name as global variables. Definitely not required for point-and-click variable inspection.
PAUSE	Stops program execution. It can be resumed later.
R / RV / RMVVAR <expression>	Removes the expression from the `WatchVar` List. You can do this by right clicking on the expression on the watchvar, too.
RL / RMVLIST	Hides the `WatchVar` List if shown
RUN	Same as F12. If the program is not in execution, you will be prompted for parameter information. Is the program is already running, the execution will continue.
STEP	Steps one executable statement. Is the same as `STEP OVER`.
STEP <number>	Steps <number> executable statements. `OVER` is assumed.
STEP OVER	When `OVER` is specified, the function or procedure calls (or OPM program calls) are considered one single statement and the debugger will not get inside them.
STEP INTO	When `INTO` is specified, the function or procedure calls (or OPM program calls) are also debugged (the debugger will jump `INTO` the calls). Debugging information is required in these modules/programs.
STEP <number> [INTO | OVER]	See above.

Command	Action
STOP	Tries to stop the running program. When debugging a batch job, means to end the debugged job. In an interactive job, it can force previous calling programs to termination too.
[SYS/SYSTEM] <system command>	Execute System Command on debugged job (`SYS` is not mandatory)
WATCH <variable or expression> [:<length>]	Allows you to stop the program at whichever statement where the contents of a specified variable (or storage location) is changed from it's current value. To monitor variable `CCODE`, enter `WATCH CCODE`. Lenght specifies how many bytes of the variable or expression are going to be monitored for changes. To stop the program whenever the first three bytes of `CUST` are changed, enter `WATCH CUST:3`. There is a maximum of active watches for the entire system. You can check the active watches with the Display Debug Watches (`DSPDBGWCH`) command.

<variable list> is one variable or more, separated by commas. Examples: `PARM01`, `ERRORCODE`, `CUSTNAME`

3.1 Note about WATCH for C/C++ programmers

It is important to understand that the watch statement establishes the watched storage location address when the watchstatement is entered, and it does not change. This can cause misleading results if a temporary storage location is watched and that storage location is reused while the application is running. An example of this is the automatic storage of an ILE C procedure, which can be reused if the procedure ends.

3.2 More information on commands

You can find more information on the `ATTR`, `BREAK`, `CLEAR`, `EVAL`, `QUAL`, `STEP`, and `WATCH` commands in the following IBM Manual: *OS/400 Debugger APIs V4R3, Document Number: SC41-5854-02, Chapter 1.34 Submit Debug Command (QteSubmitDebugCommand) API*, as the syntax is equivalent to that of the IBM ILE debugging API described in there.

Chapter 4

How to use Expressions in VISUAL Message Center Debugging Agent

Expressions (in the command reference, they are mentioned as `<expression>`) are language dependant. You need to know how to formulate an expression for entering a conditional statement, for instance, or to evaluate it. In most cases, however, you are not going to need to type an expression, as the point and click features of VISUAL Message Center Debugging Agent will save you from that.

The following expression:

```
%SUBSTR(&CUSTCODE 2 1) <= X'C1'
```

is valid when debugging a CL program (if returns true or false, and can be used to set a conditional breakpoint to see if the second byte of `&CUSTCODE` is less than or equal to hexadecimal value C1. The expression:

```
ARRAYC(4) = 3.14
```

can be used when debugging an RPG program in an `EVAL` or conditional breakpoint. It tests the fourth element on `ARRAYC` to be equal to the decimal value `3.14`. You can use functions in the expression, as in:

```
%SUBSTR(CHAR10  7  4) = 'TEXT'
```

You can reference RPG indicators using parenthesis or not, as in `*IN03` and `*IN(3)`. However, `*IN,3` is not valid.

In COBOL, you can use the `OF` clause to qualify variables, as in:

```
CUST-CODE OF CUSTOMER-RECORD
```

In C or C++ you can use the `*` in order to reference the storage area pointed by a pointer variable, as in:

```
*pointerXX
```

where `pointerXX` is a pointer. You can use the :s format option (see below for more information on formatting option codes) to see the storage pointed by `pointerXX` in string format:

```
EVAL *pointerXX:s
```

You can also reference multiple array element by indicating an interval. For instance:

```
ARRAYC(3452)
```

means the elements `34` to `52` in the array named `ARRAYC`.

4.1 The %VAR function

If the variable has the same name as a debug command, the `%VAR` keyword must be used.

For instance, to evaluate the RPG variable `EVAL` enter the command: `EVAL %VAR(EVAL)`.

Chapter 5
EVAL Command Formatting option codes

Formatting option codes help you formatting the output from an EVAL command, just in case you dislike the default display presentation. Usually you are not going to use them unless you are using pointers. *(Note for experienced programmers: the assumed code page of the string is also given. This CCSID is converted to the job CCSID. If 65535 is shown, no conversion is done.)*

5.1 For RPG, CL and all languages:

Command	Description
:c <length>	Character formatting - Default length is 1. EBCDIC CCSID 65535.
:x <length>	Hexadecimal formatting - Default length is the length of the expression value. EBCDIC CCSID 65535.

5.2 For all ILE languages:

(For instance, ILE RPG but NOT RPG/400) and Java the following additional formatting code options are supported:

Command	Description
:a <length>	String formatting - Default length is 1024. ASCII related job CCSID.
:u <length>	String formatting - Default length is 1024. Unicode CCSID 13488.

5.3 For C/C++:

The following additional formatting code options are supported:

Command	Description
:s <length>	String formatting - Default length is 30. EBCDIC CCSID 65535 (supported for Java also).
:f <length>	String newline formatting - Default length is 1024. EBCDIC CCSID 65535.

Chapter 6
Summary by Activity

6.1 Searching

Action	Shortcut
Find	Ctrl+F
Find Next	Shift+F4 (F16)
FIND String Command	F/FIND <string> N/NEXT | P/PREV | F/FIRST | L/LAST | A/ALL
FIND Extended Command	F/FIND <*BKP | *BKPNOEXE | *CHG | *COM | *NOCOM> N/NEXT | P/PREV | F/FIRST | L/LAST | A/ALL

6.2 Positioning

Action	Shortcut
Go top Command	T/TOP
Go Bottom Command	B/BOT/BOTTOM
Go to (Add) module/program	Left Mouse Click (when the cursor is located over an hyperlinked module or program name)
Go to Procedure/Subroutine Definition	Letf Mouse Click (when the cursor is located over an hyperlinked procedure or subroutine name)
Back	Ctrl+B (after hyperlink jump)
Next (Forward)	Ctrl+N (after going back from hyperlink jump)
See Other Loaded Module	F7 (switches to next one cyclically)
Go to Next Breakpoint	Shift+F8 (F20)
Go to Execution Point	Ctrl+G
Go to (Add) module/program	Left Mouse Click (on module names in Program Navigator)
Go to Procedure/SubRt/Label Definition	Left Mouse Click (on label/name in Program Navigator)

Action	Shortcut
Go to Reference	Left Mouse Click (on any references in Program Navigator)
Go to top of source	Ctrl+Home
Go to end of source	Ctrl+End
Go to beginning of source line	Home
Go to end of source line	End
By-word move	Ctrl+left/right arrow keys
By character move	Left/right arrow keys
By line move	Up/Down arrow keys
By Page move	PgDown/PgUp keys

6.3 Setting / Removing Breakpoints

Action	Shortcut
Set/Clear Breakpoint	F6 (when the cursor is located at a breakable statement) or clicking the Set/Clear Breakpoint icon in toolbar
Set/Clear Conditional Breakpoint	Shift+F6 (F18) (when the cursor is located at a breakable statement) or clicking the Set/Clear Conditional Breakpoint icon in toolbar
Set/Clear Temporal Breakpoint	Ctrl+F6 (when the cursor is located at a breakable statement) or clicking the Set/Clear Temporal Breakpoint icon in toolbar
Setting multiple breakpoints	Highlight many statements (by moving the cursor and pressing Shift at the same time) then click on the Breakpoint or Conditional breakpoint icons in the toolbar. By using the Contextual Menu Right Mouse Click (when the cursor is located at a breakable statement)
Toggle Breakpoint On/Off	Left Mouse Click (on blue dot or breakpoint indication at the Line Status Area)
Set/Clear Conditional Breakpoint	Shift+Left Mouse Click (on blue dot or breakpoint indication at the Line Status Area)
Add All Breakpoints	Shift+F10 (F22)
Remove ALL Breakpoints	Shift+F11 (F23)
Set a breakpoint using Commands	BREAK / AT <statement list> WHEN <condition>
Remove a breakpoint using Commands	CLEAR <statement list>
Remove all breakpoints using Commands	CLEAR PGM
Run to Cursor	F4

6.4 Showing variable values

Action	Shortcut
Show / Change Variables	F2 or Work with Variables icon in toolbar when a variable name has been highlighted or when a variable is selected in Data Dictionary
Show Tooltip Variable	Right Mouse Click when a variable name has been highlighted or when in Data Dictionary
Show variables using the WatchVar list	Use the watch icon or the commands AV/ADDVAR/RV/RMVVAR or right click on the watch list to get a context menu

6.5 Controlling execution

Action	Shortcut
Step Into	F8 or STEP <number> INTO command or STEP INTO icon in toolbar
Step Over	F9 or STEP <number> OVER command or STEP OVER icon in toolbar
Run / Resume Execution	F12 or RUN command or RUN icon in toolbar
Stop Execution	Press F3 twice or use the STOP command or the STOP icon in toolbar
Pause Execution	Press F3 or PAUSE icon in toolbar

6.6 Monitoring variables

Action	Shortcut
Monitor a variable or some bytes of it	Use the WATCH <variable or expression> [:<length>] command. Example: WATCH CUST:3 stops whenever the first three bytes of CUST are changed. WATCH CCODE stops wherever CCODE changes.

Remember you can capture the job at any time to see the actual end-user screen (up to the last keystroke), the LDA, the QTEMP, the Call Stack. you only need to press Ctrl-J (Open Job) or the Open Job icon in the toolbar.

Appendix A
Contacting Tango/04

North America

Tango/04 North America
PO BOX 3301
NH 03458 Peterborough
USA

Phone: 1-800-304-6872 / 603-924-7391
Fax: 858-428-2864
sales@tango04.net
www.tango04.com

EMEA

Tango/04 Computing Group S.L.
Avda. Meridiana 358, 5 A-B
08027 Barcelona
Spain

Phone: +34 93 274 0051
Fax: +34 93 345 1329
info@tango04.net
www.tango04.com

Italy

Tango/04 Italy
Viale Garibaldi 51/53
13100 Vercelli
Italy

Phone: +39 0161 56922
Fax: +39 0161 259277
info@tango04.it
www.tango04.it

Sales Office in France

Tango/04 France
La Grande Arche
Paroi Nord 15ème étage
92044 Paris La Défense
France

Phone: +33 01 40 90 34 49
Fax: +33 01 40 90 31 01
contact@tango04.net
www.tango04.fr

Sales Office in Switzerland

Tango/04 Switzerland
18, Avenue Louis Casaï
CH-1209 Genève
Switzerland

Phone: +41 (0)22 747 7866
Fax: +41 (0)22 747 7999
contact@tango04.net
www.tango04.fr

Latin American Headquarters

Barcelona/04 Computing Group SRL (Argentina)
Avda. Federico Lacroze 2252, Piso 6
1426 Buenos Aires Capital Federal
Argentina

Phone: +54 11 4774-0112
Fax: +54 11 4773-9163
info@barcelona04.net
www.barcelona04.com

Sales Office in Peru

Barcelona/04 PERÚ
Centro Empresarial Real
Av. Víctor A. Belaúnde 147, Vía Principal 140
Edificio Real Seis, Piso 6
L 27 Lima
Perú

Phone: +51 1 211-2690
Fax: +51 1 211-2526
info@barcelona04.net
www.barcelona04.com

Sales Office in Chile

Barcelona/04 Chile
Nueva de Lyon 096 Oficina 702,
Providencia
Santiago
Chile

Phone: +56 2 234-0898
Fax: +56 2 2340865
info@barcelona04.net
www.barcelona04.com

About Tango/04 Computing Group

Tango/04 Computing Group is one of the leading developers of systems management and automation software. Tango/04 software helps companies maintain the operating health of all their business processes, improve service levels, increase productivity, and reduce costs through intelligent management of their IT infrastructure.

Founded in 1991 in Barcelona, Spain, Tango/04 is an IBM Business Partner and a key member of IBM's Autonomic Computing initiative. Tango/04 has more than a thousand customers who are served by over 35 authorized Business Partners around the world.

Alliances

Partnerships

IBM Business Partner

IBM Autonomic Computing Business Partner

IBM PartnerWorld for Developers Advanced Membership

IBM ISV Advantage Agreement

IBM Early code release

IBM Direct Technical Liaison

Microsoft Developer Network

Microsoft Early Code Release

Awards

Legal Notice

The information in this document was created using certain specific equipment and environments, and it is limited in application to those specific hardware and software products and version and releases levels.

Any references in this document regarding Tango/04 Computing Group products, software or services do not mean that Tango/04 Computing Group intends to make these available in all countries in which Tango/04 Computing Group operates. Any reference to a Tango/04 Computing Group product, software, or service may be used. Any functionally equivalent product that does not infringe any of Tango/04 Computing Group's intellectual property rights may be used instead of the Tango/04 Computing Group product, software or service

Tango/04 Computing Group may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

The information contained in this document has not been submitted to any formal Tango/04 Computing Group test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility, and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Despite the fact that Tango/04 Computing Group could have reviewed each item for accurateness in a specific situation, there is no guarantee that the same or similar results will be obtained somewhere else. Customers attempting to adapt these techniques to their own environments do so at their own risk. Tango/04 Computing Group shall not be liable for any damages arising out of your use of the techniques depicted on this document, even if they have been advised of the possibility of such damages. This document could contain technical inaccuracies or typographical errors.

Any pointers in this publication to external web sites are provided for your convenience only and do not, in any manner, serve as an endorsement of these web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries: iSeries, iSeriese, iSeries, i5, DB2, e (logo)®Server IBM ®, Operating System/400, OS/400, i5/OS.

Microsoft, SQL Server, Windows, Windows NT, Windows XP and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries. UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group. Oracle is a registered trade mark of Oracle Corporation.

Other company, product, and service names may be trademarks or service marks of other companies.