

WHITE PAPER | APPLICATION SERVICES

# NTT DATA Mainframe Re-hosting Development Environment

JUNE 2018



A critical aspect of any mainframe migration is to ensure the productivity of the development team is maintained during and after the transition to the newly re-hosted environment. This can be achieved by providing a selection of NTT DATA and third-party software solutions that aim to replicate the mainframe development environment as close as possible.

On many occasions, the newly introduced solutions improve productivity. Because there's no change to the underlying application or business process, the responsibilities of each development team member will be retained, while increasing productivity.

A common goal when planning migrations is to align with specific distributed development standards and, if available, published reference architectures. This allows organizations to consolidate solutions and skill sets, while aligning the migration with the future direction of the organization.

This paper provides an overview of development solutions — based on NTT DATA and third-party-provided open systems — that aim to replicate the mainframe development environment.

Key areas include:

- **Application development environment:** This section focuses on the available target solutions that support online and batch-based application development and maintenance. Solutions discussed include an NTT DATA-supplied COBOL integrated development environment (IDE) and a distributed version of the common mainframe Interactive System Productivity Facility (ISPF)-based solution.
- **Data management:** NTT DATA's Transaction Processing Environment (TPE) and Batch Processing Environment (BPE) support multiple mainframe data types, including virtual storage access method (VSAM), sequential and relational-based data. This section focuses on the solutions available to manage the different data types within the distributed environment.
- **Application debugging:** It's imperative that application developers are able to debug and resolve application problems during unit testing. This section covers available solutions, ranging from graphical source debuggers to supplied command line utilities.
- **Source code control and build management:** A common goal of any mainframe re-hosting project is to consolidate solutions and standardize on previously established processes whenever possible. On many occasions, organizations already have software lifecycle-related distributed development standards that can be utilized by the newly re-hosted applications. This section discusses the integration with distributed source code and integrated build management solutions.

## Application development environment

Application development environment requirements differ. Some developers want to take advantage of new IDE capabilities, whereas others would rather utilize a familiar mainframe-based development environment. NTT DATA offers a variety of options (or combination of options) for mainframe application developers who are transitioning to a distributed environment.

### Option 1: An integrated development environment

Enterprise COBOL IDE by NTT DATA offers the choice of an IDE based on Microsoft Visual Studio or Eclipse. This allows users to develop and debug re-hosted Customer Information Control System (CICS), Information Management System (IMS), batch-based COBOL applications, and batch Job Control Language (JCL), as well as to maintain and extend screens based on CICS Basic Mapping Support (BMS) and IMS Message Format Service (MFS) from within the IDE.

Key features of Enterprise COBOL IDE include:

- Support for the standard (integrated) version of Visual Studio and the isolated shell version of Visual Studio (via a separate, self-contained distributable), in addition to support for an Eclipse-based application workbench
- A built-in precompiler for TPE (CICS and IMS TM) pre-processing

- JCL editing and integration with BPE for remote translation, job submission and batch debugging
- Screen editing for screen definitions based on both CICS BMS and IMS MFS
- Built-in source-level debugging from within the IDE; the ability to set breakpoints and watch specific values (in either ASCII or HEX view)
- Color coding for COBOL keyword, statements, strings and other language elements
- Automatic syntax error detection as you edit source code using the Visual Studio IDE; errors are highlighted in the source code editor using a red squiggle underline, which you can hover over to display a description of the error in a tooltip window
- Remote debugging capabilities for applications deployed on non-Windows-based platforms such as Linux or UNIX
- Integrated FTP/SFTP deployment to remote platforms for test and/or release
- Integration with source code management solutions such as Microsoft Team Foundation Server and Subversion

While COBOL source can be compiled locally within the Enterprise COBOL IDE, there will be occasions when the source will need to be compiled on the target platform, such as Linux or UNIX; for example, for database pre-processing using the Oracle Pro\*COBOL pre-processor.

To compile programs remotely, use Enterprise COBOL Remote Compiler by NTT DATA, which consists of a server listener that receives source files from the clients, precompiles and compiles them locally on the server, and eventually sends the resulting classes and translated source files back to the clients. The net result is that the developer doesn't have to leave Enterprise COBOL IDE.

Figure 1 shows the main source code editing window for the Visual Studio version of Enterprise COBOL IDE.

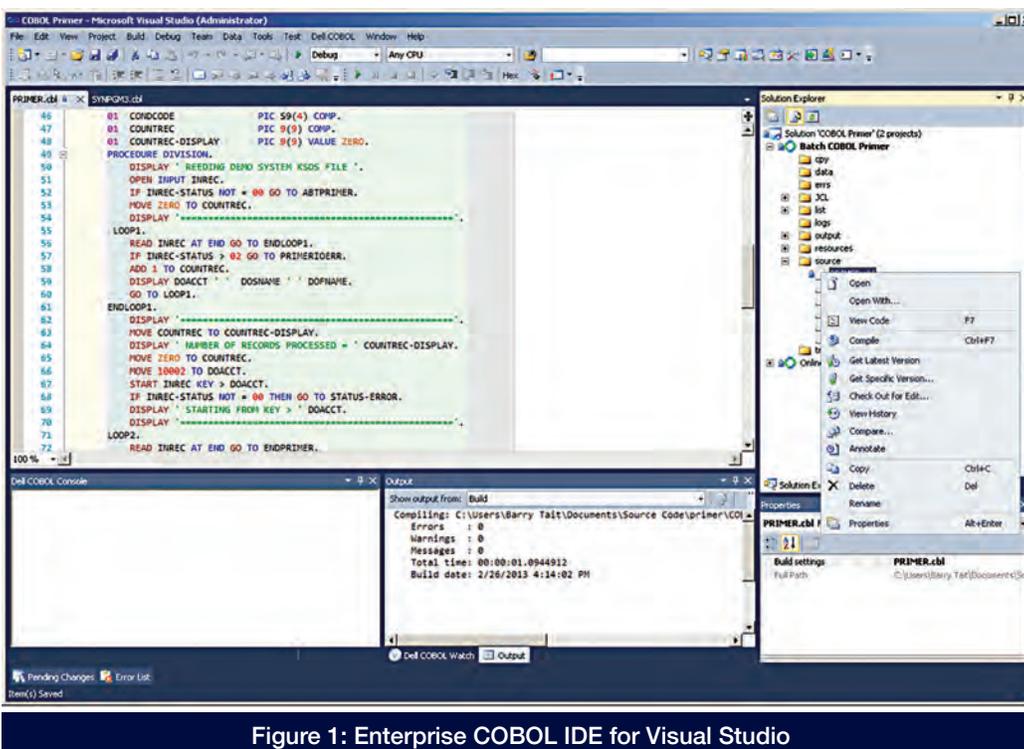


Figure 1: Enterprise COBOL IDE for Visual Studio

### Remote deployment using Enterprise COBOL IDE

Files can be deployed to the remote server for testing using the Enterprise COBOL Deploy Editor by NTT DATA. You will be required to configure one or more remote servers prior to deploying. Figure 2 shows the Enterprise COBOL Deploy Editor window.

Enterprise COBOL IDE is provided as part of the NTT DATA Enterprise COBOL Suite. Additional features include a streamlined, portable runtime environment, a powerful COBOL compiler, and a range of data access and migration utilities to simplify transitions. The COBOL compiler translates COBOL source code into Java byte code objects that are then executed with the Java Virtual Machine (JVM), which produces 100% portable runtime code that can be deployed on a variety of platforms.

### Option 2: A familiar ISPF-based development environment

NTT DATA also offers an alternative development environment that gives developers the choice to utilize an ISPF-based development environment. With uni-SPF Extended for TPE/BPE by NTT DATA, developers can transition to a distributed system while continuing to utilize a familiar mainframe-based ISPF environment for many of their common development-related tasks, including:

- COBOL source compilation with support for embedded database and CICS calls. Mass compilation is enabled through the use of a wild card. Program source can be edited, and compiled results can be viewed directly from the compilation panel. Because the compilation panel acts like a console, the user never has to leave the panel to edit compile or check results; there is no need to even split the screen.
- File editing and browsing that is nearly identical to the mainframe. Support is

provided for most of the primary line commands used in ISPF; for example, CREATE, REPLACE, MM, CC, and both the )) and >> forms of shift.

- Integration with Mainframe Re-hosting software by NTT DATA providing users with an environment to develop, manage and maintain batch jobs. Users have the ability to submit the original JCL source, which will then be translated automatically at runtime. Users can also check the status of TPE regions, access log files, launch a terminal and perform other TPE-related activities.
- Ad hoc batch job submission, which is possible from within an ISPF edit session using a standard submit (sub) command. Once running, job status can be queried via the integrated BPE active jobs command, and job output browsed – all from within the uni-SPF Extended session.
- SDSF capabilities for browsing job output. uni-SPF Extended for provides the ability to show a job's SYSOUT data sets, job log, the JCL for the job and any job-related messages. Any of the outputs can be optionally printed and searched.
- The ability for authenticated/privileged users to send email and trigger SFTP services directly from the uni-SPF Extended panel.
- A provision that exists to create a menu of FastPath access for frequently used files. For the developer, an extended cut-and-paste feature is also included,

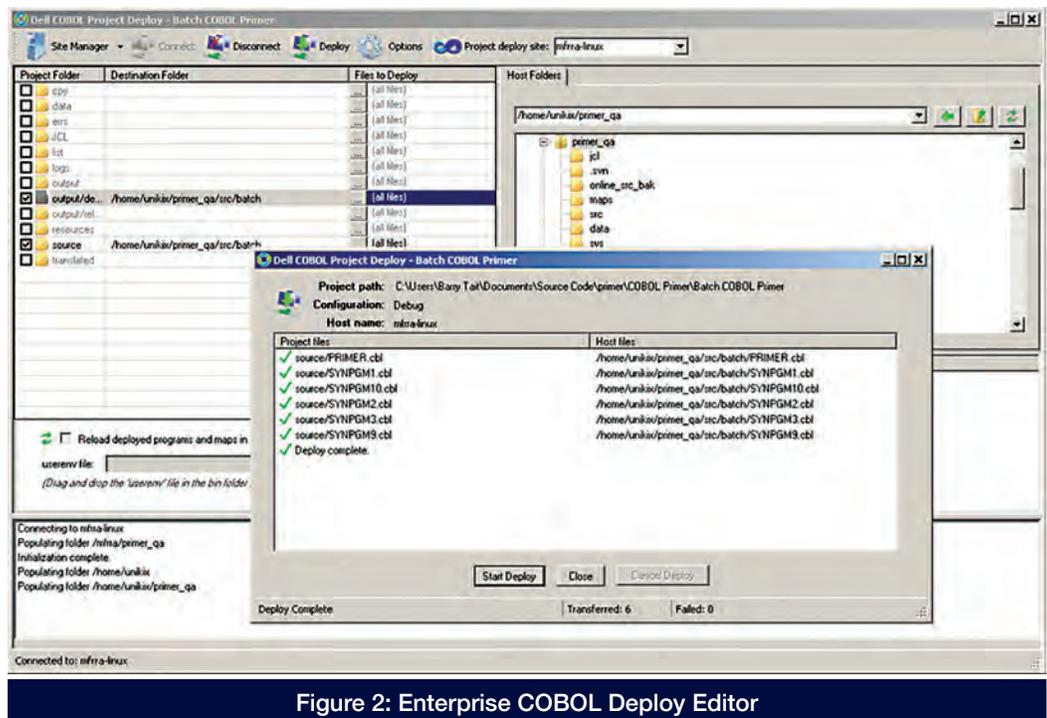


Figure 2: Enterprise COBOL Deploy Editor

which has the ability to maintain up to 99 boilerplate pieces of code that are available to paste in virtually any source file. This extended cut-and-paste feature can also be used to keep notes, if desired.

- Integration with common distributed source code management solutions such as Subversion and Microsoft Team Foundation Server from customized ISPF panels.

In addition, custom panels and related scripting can be added to the product in order to replicate existing client functionality available today.

### Option 3: Linux/UNIX-based command line utilities

NTT DATA also offers the ability for developers to execute TPE, BPE and Enterprise COBOL-related utilities directly on the Linux/UNIX-based platform. There are several Linux/UNIX-based commands and utilities that will become invaluable to the developer, such as the diff command for comparing two files and ps for reporting the status of a process. Developers will also have the ability to use custom scripting in order to automate many development tasks.

Note that although the use of a Windows-based solution like Enterprise COBOL IDE will help developers avoid the distributed platform, it will be extremely difficult to avoid at least some interaction with the target Linux/UNIX command line. To address this issue, NTT DATA provides a Linux/UNIX fundamentals training course that aims to equip each developer with enough basic skills to navigate comfortably around the Linux/UNIX file system.

## Data management

For file management, there's the NTT DATA File Editor Suite, which consists of two main components. File Editor by NTT DATA is an Eclipse-based application that replaces the existing capabilities of mainframe-based data editing solutions such as File-AID and File Manager. File Access Server Agent by NTT DATA is installed on the system where the data file resides. In addition, uni-SPF Extended provides the ability to view and modify fixed-length sequential data.

### Database management

There are several target database development tools available. Oracle SQL Developer is a free IDE that simplifies the development and management of the Oracle Database. It offers complete end-to-end development of PL/SQL applications, a worksheet for running queries and scripts, a database administrator console, a reports interface and a complete data modeling solution.

IBM DB2 LUW comes with a free SQL development environment named IBM Data Studio. Data Studio provides a simple tree-based model of the physical implementation of the database objects and definitions. This enhances developer productivity without requiring in-depth knowledge of the database layout, as the IDE-style GUI environment will be familiar to developers and database administrators. Additionally, Data Studio provides a quick and easy means for performing ad hoc queries on various database objects. These queries can be built using a wizard-type interface or interactively using pop-up guides as the statements are entered. Once complete, the SQL query can be executed and the result set immediately displayed. At any time, the query code and the result set can be saved locally for later analysis.

```

EDIT /home/unikix/transam/jcl/jmvs/TST04MFR -----
COMMAND ==> aut
JCL member
000001 //TST04MFR JOB JF21,MSGCLASS=J,CLASS=A,
000002 //      REGION=4M
000003 //JOBLIB DD DISP=SHR,DSN=TFHT.GEN1.TEMP.LOADLIB
000004 //      DD DISP=SHR,DSN=TFIPROD.BATCH.LOADLIB
000005 //      DD DISP=SHR,DSN=TFHT.LOADLIB
000006 /**
000007 /**
000008 //STEP1 EXEC ISTD3PRC,PARM='FOC001',COND=EVEN,REGION=2048K
000009 /**
000010 /**
000011 /**
000012 //STEP2 EXEC PGM=IEFBR14
000013 /**
000014 //TSD01 DD DSN=ISDPR.DML.BACKUP,DISP=(SHR,DELETE,KEEP)
000015 /**
***** BOTTOM OF DATA *****

```

Figure 3: uni-SPF Extended job submission screen

```

Job History /mfrfa/transam/history -----
COMMAND ==>
Valid options: S=Select, ?=Display Datasets
Filter ==> TST04MFR* e.g., *, A, A*, AB*, ut* etc, Case sensitive!

```

S	Name	Stat	Perms	Date Modified	Size	Owner
█	TST04MFR_217.hst.2014-11-13.11.59	0664	Nov 13 11:59	30101	unikix	
-	TST04MFR_213.hst.2014-11-04.10.02	0664	Nov 04 10:02	30099	unikix	
-	TST04MFR_199.hst.2014-10-21.12.34	0664	Oct 21 12:35	30101	unikix	
-	TST04MFR_187.hst.2014-09-17.08.18	0664	Sep 17 08:18	30007	unikix	
-	TST04MFR_186.hst.2014-09-17.08.18	0664	Sep 17 08:18	30007	unikix	
-	TST04MFR_185.hst.2014-09-17.08.18	0664	Sep 17 08:18	30007	unikix	
-	TST04MFR_184.hst.2014-09-17.08.18	0664	Sep 17 08:18	30007	unikix	
-	TST04MFR_175.hst.2014-09-16.03.40	0664	Sep 16 03:41	30101	unikix	
-	TST04MFR_174.hst.2014-09-16.03.40	0664	Sep 16 03:41	30101	unikix	
-	TST04MFR_169.hst.2014-09-12.15.27	0664	Sep 12 15:27	30001	unikix	
-	TST04MFR_154.hst.2014-08-20.11.52	0664	Aug 20 11:53	30437	unikix	

Figure 4: uni-SPF Extended job output screen

## Application debugging

The Enterprise COBOL IDE includes a sophisticated integrated debugger, allowing developers to perform source-level debugging activities against applications that have been deployed to remote platforms. Activities that are supported include stepping through COBOL source code, setting breakpoints and monitoring specific variables. The debugger can be invoked from Enterprise COBOL IDE while locally editing a job or executing an online TPE transaction via a terminal emulator.

### Online application debugging and testing

In addition to the debugger, TPE provides different utilities to aid in the debugging and resolution of transaction and system abnormal ends. TPE also provides log files and dumps that can be interrogated to obtain critical program runtime information such as the EXEC interface block (EIB), Transaction Work Area (TWA), Common Work Area (CWA), storage area and working storage. A description of some of the utilities available to developers can be seen in Table 1.

Utility name	Description
kixdump	The kixdump utility displays information about the operation of a specific TPE region. It can show the transactions that are currently executing, table and file information, the trace table and statistical information. This program links to the region's shared memory areas and can run only when these shared memory areas are present. If shared memory is corrupt, the output from kixdump contains information about the cause of the corruption.
kixcontrol	The kixcontrol utility is used to dynamically turn dumps and traces on and off. It provides options that also enable you to limit the scope of the trace to a transaction, terminal, program, data set or command. The specialized trace options are designed to be used with the guidance of support personnel to diagnose problems.
kixdate	The kixdate date/time configuration utility provides a menu-driven interface to change the date/time for TPE and BPE systems (replacing functionality provided by a solution such as CA Validate). By default, TPE and BPE reflect the system time. All dates displayed by TPE in its logs, screens, traces, dumps and reports use four digits for the year.
kixsnap	The TPE-supplied kixsnap utility captures information about a region's transaction processors and tables, system processes and users at the time of an abnormal termination. kixsnap creates a compressed file that can be sent directly to NTT DATA support for further investigation. When a system crash occurs, the kixsnap utility is executed automatically.
CEDF	Command Execution Diagnostic Facility (CEDF) is a CICS-supplied transaction that provides source-level CICS command debugging capabilities. CEDF provides a wide range of options for setting breakpoints — locations where control is interrupted and passed to the transaction server for debugging purposes. CEDF allows debugging for single or multiple terminal sessions, as well as debugging for routed transactions.
CEBR	The CICS-supplied Execution Browse (CEBR) transaction is used to browse temporary storage and to retrieve and insert data into intrapartition queues. The kixdump-q command can also be used to obtain a list of temporary storage queues currently in a region, which the NTT DATA TPE Manager can also list.
CEMT	The CICS-supplied Extended Master Terminal (CEMT) transaction is used to perform system maintenance tasks, including: <ul style="list-style-type: none"> <li>Viewing the current status of foreground and background tasks</li> <li>Viewing transaction classes and changing the number of transaction processors assigned to a transaction class</li> <li>Loading a program, mapset or shared library</li> <li>Purging a specific task or all tasks (useful if a task goes into a loop)</li> <li>Putting into and taking out of service 3270 terminals and connections to remote systems</li> <li>Changing the state of a transaction</li> <li>Changing the system state</li> <li>Shutting down the region</li> </ul>
TPE trace facility	The TPE trace facility enables users to selectively trace low-level routines. The trace facility should be used only under the direction of a technical support engineer. This internal tracing facility increases the amount of shared memory that TPE allocates.
TPE log files	The unikixmain.log file contains information about the transaction failure. It provides the name of the transaction, the name of the program and the last four CICS commands executed by the transaction. Each CICS command provides: <ul style="list-style-type: none"> <li>The name of the CICS command executed</li> <li>CID# references, which stand for CICS command identification reference numbers</li> <li>Status of the command executed</li> </ul> The CID# reference helps locate the CICS command(s) that was executed. This reference number is generated by the kixclt translator. By searching for the CID# reference in the output of the kixclt translator (a *.cbl file for COBOL), you can locate the CICS command that was executed.

Table 1: Available TPE-related debugging utilities

## Batch application debugging and testing

In addition to the debugger, BPE provides many different utilities that help debug and resolve batch-related issues, as shown in Table 2.

Utility name	Description
anmjob	Invokes the debugger for a batch job.
abtjob	Cancels a job or a command that is currently executing.
chgjob	Changes the attributes of the specified job.
ebmsnap	Creates a system snapshot when troubleshooting a problem. Snapshots can be sent to NTT DATA support for further investigation.
infjob	Provides information about a currently executing job.
insjbl	Helps determine the status of one or more waiting jobs, during and after it completes.
insjob	Helps obtain the status of a job that is submitted, executing or terminated.
lgprint	Extracts the BPE internal trace entries from the trace files defined by the admlog command and writes this information to the \$PUBLIC/msg/elog_print file.

Table 2: Available BPE utilities for application debugging

In addition to the utilities listed in Table 2, the Batch Administration Manager (BAM) problem determination menu (option 6 from the main menu) contains many different options which aid in determining the root cause of a problem, such as access to the error log file, batch system tracing and troubleshooting. The latter allows a user to run a sample test against the batch subsystem in order to determine if the subsystem is functioning correctly.

## Source code control and build management

### Source code management

As with the majority of third-party solutions, NTT DATA is agnostic in regard to the selection of a source code management solution. NTT DATA has experience with Subversion, Git, Microsoft Team Foundation Server, PVCS Version Manager and Concurrent Versions System (CVS).

It is imperative that any new development environment has the ability to integrate with the selected source code management solution. The two main development environments mentioned in this paper — Enterprise COBOL IDE (Visual Studio- and Eclipse-based versions) and an ISPF-based environment for developers who are looking to utilize a recognized mainframe look and feel — can integrate with leading source control solutions, such as Subversion and Microsoft Team Foundation Server, via standard plug-ins.

### Automated build management

NTT DATA is also agnostic regarding the selection of an automated build management solution, but has experience with Jenkins,\* a continuous build and integration solution. Jenkins is part of Mainframe Re-hosting Reference Architecture by NTT DATA, where it is pre-configured to perform quality assurance (QA) builds whenever a developer checks an asset back into the Subversion repository from the Enterprise COBOL IDE following unit testing. The relevant source is then extracted on the target Linux QA server, compiled using a specific build script and then compressed into a zip-based build distributable file. This file is then transferred to another Linux-based test environment, where it is extracted and ready for system tests.

## Recommended developer training

NTT DATA offers several training courses to aid developers with their transition to the new distributed development environment. The Linux/UNIX fundamentals course is ideal for developers with little to no existing distributed experience. The TPE, BPE and Enterprise COBOL training courses are mandatory for all developers without previous experience in that environment. There is also an optional uni-SPF Extended training class for any developers interested in this solution.

### Linux/UNIX fundamentals

NTT DATA offers a half-day Linux crash course specifically focused on teaching mainframe-skilled personnel the basic commands required to survive in the Linux/UNIX world. The benefit of this class is that it's focused on teaching students distributed skills that are relevant to the NTT DATA re-hosting environment. The course can be delivered as part of the TPE/BPE product training (typically the first day, as basic Linux/UNIX skills are a prerequisite of the TPE/BPE product training).

During the course, students will learn the following:

- Linux/UNIX-based terminology, system components, shell functions, case sensitivity, user IDs, groups and environment variables
- File system commands, directory structures and navigation, file permissions, moving and copying files and directories, removing files and creating and removing directories
- Linux/UNIX tools (such as man, more, date, who, echo, cat), redirecting output, examining files and variables, searching files (grep) and monitoring processes
- Shell scripting: creating shell variables, return status, if-then-else construct, case construct and for loops
- vi Editor\*: operating modes, inserting and deleting text, copying and moving content, searching for strings, and moving around in a file
- Common vulnerabilities and security risks associated with the Linux/UNIX operating environment

Even though NTT DATA recommends uni-SPF Extended for TPE/BPE, it would make sense for all developers to have some visibility to the vi editing environment, as it is common across all Linux/UNIX systems.

### TPE/BPE product development and administration training

NTT DATA offers two product-related training courses:

- **NTT DATA TPE Software Development and Administration** provides students with the essential information and skills to develop and maintain online-based applications within TPE. Students will also learn to install, customize and maintain the TPE software and get an introduction to the TPE architecture, TPE transaction processor building, TPE region creation, BMS translation, and online program translation, compilation, execution and debugging.
- **BPE Software Development and Administration** provides students with the essential information and skills to develop and maintain batch-based applications within BPE. Students will also learn to install, customize and maintain the BPE software and get an introduction to BPE architecture, BPE subsystem creation, job and procedure translation, batch program compilation and BPE batch job submission.

### Enterprise COBOL training

The courses (listed above) also have application development modules covering the following areas:

- Enterprise COBOL IDE
- Enterprise COBOL-related environment variables
- Enterprise COBOL compilation options
- Enterprise COBOL output files
- Enterprise COBOL program preparation for source-level debugging using the Enterprise COBOL debugger
- Application debugging using the Enterprise COBOL debugger

\*Jenkins — An extensible open source continuous integration server, <http://ankhsvn.open.collab.net/>

## Summary

When moving from one platform to another, there is always an initial learning curve as developers transition to new tools within a new environment on a new platform. To ease this transition, NTT DATA offers standard product training around many of the distributed development solutions described in this paper.

Mainframe developers transitioning to the new distributed environment will have the opportunity to utilize a familiar ISPF-based development environment and/or use a new Visual Studio-based integrated development environment. Once past the initial learning curve, the additional available features will aid in increasing the overall productivity of the developer.

Visit [nttdataservices.com](http://nttdataservices.com) to learn more.

NTT DATA Services partners with clients to navigate and simplify the modern complexities of business and technology, delivering the insights, solutions and outcomes that matter most. As a division of NTT DATA Corporation, a top 10 global IT services and consulting provider, we wrap deep industry expertise around a comprehensive portfolio of infrastructure, applications and business process services.

**NTT DATA**