

Micro Focus[®]

Modernization Workbench[™]

Batch Refresh Process



Copyright © 2009 Micro Focus (IP) Ltd. All rights reserved.

Micro Focus (IP) Ltd. has made every effort to ensure that this book is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time. The software described in this document is supplied under a license and may be used or copied only in accordance with the terms of such license, and in particular any warranty of fitness of Micro Focus software products for any particular purpose is expressly excluded and in no event will Micro Focus be liable for any consequential loss.

Micro Focus, the Micro Focus Logo, Micro Focus Server, Micro Focus Studio, Net Express, Net Express Academic Edition, Net Express Personal Edition, Server Express, Mainframe Express, Animator, Application Server, AppMaster Builder, APS, Data Express, Enterprise Server, Enterprise View, EnterpriseLink, Object COBOL Developer Suite, Revolve, Revolve Enterprise Edition, SOA Express, Unlocking the Value of Legacy, and XDB are trademarks or registered trademarks of Micro Focus (IP) Limited in the United Kingdom, the United States and other countries.

IBM®, CICS® and RACF® are registered trademarks, and IMS™ is a trademark, of International Business Machines Corporation.

Copyrights for third party software used in the product:

- The YGrep Search Engine is Copyright (c) 1992-2004 Yves Roumazeilles
- Apache web site (<http://www.microfocus.com/docs/links.asp?mfe=apache>)
- Eclipse (<http://www.microfocus.com/docs/links.asp?nx=eclp>)
- Cyrus SASL license
- Open LDAP license

All other trademarks are the property of their respective owners.

No part of this publication, with the exception of the software product user documentation contained on a CD-ROM, may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine-readable form without prior written consent of Micro Focus (IP) Ltd. Contact your Micro Focus representative if you require access to the modified Apache Software Foundation source files.

Licensees may duplicate the software product user documentation contained on a CD-ROM, but only to the extent necessary to support the users authorized access to the software under the license agreement. Any reproduction of the documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

U.S. GOVERNMENT RESTRICTED RIGHTS. It is acknowledged that the Software and the Documentation were developed at private expense, that no part is in the public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at FAR 52.227-19, as applicable. Contractor is Micro Focus (IP) Ltd, 9420 Key West Avenue, Rockville, Maryland 20850. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.

Contents

Preface

<i>Audience</i>	v
<i>Organization</i>	vi
<i>Conventions</i>	vi
<i>Related Manuals</i>	vi
<i>Online Help</i>	vii

1 Using the Batch Refresh Process

<i>Understanding the Batch Refresh Process</i>	1-1
<i>Configuring the Batch Refresh Process</i>	1-2
<i>Configuring General Settings</i>	1-4
<i>Configuring User Exits</i>	1-6
<i>Configuring Advanced Settings</i>	1-9
<i>Running the Batch Refresh Process</i>	1-11
<i>Preparation</i>	1-11
<i>Execution</i>	1-11
<i>Optional Processing</i>	1-12
<i>BRP Logging</i>	1-13

<i>Producing Utilities for BRP</i>	1-14
<i>Versioning</i>	1-14
<i>Logging</i>	1-15
<i>Source, Executable, and CFG Files</i>	1-15
<i>Guidelines for BRP Utilities</i>	1-15
<i>BRP and Non-BRP Modes</i>	1-16
<i>Using User Exits</i>	1-16
<i>Parameter Data</i>	1-16
<i>Logging</i>	1-17
<i>Input/Output</i>	1-17
<i>Returning Values</i>	1-18
<i>BRP Environment Parameters</i>	1-18
<i>Testing</i>	1-25

Index

Preface

The Modernization Workbench is a suite of PC-based software products for analyzing, re-architecting, and transforming legacy applications. The products are deployed in an integrated environment with access to a common repository of program objects. Language-specific parsers generate repository models that serve as the basis for a rich set of diagrams, reports, and other documentation.

The Modernization Workbench suite consists of customizable modules that together address the needs of organizations at every stage of legacy application evolution: maintenance/enhancement, renovation, and modernization.

Audience

This guide assumes that you are a corporate Information Technology (IT) professional with a working knowledge of the legacy platforms you are using the product to analyze. If you are transforming a legacy application, you should also have a working knowledge of the target platform.

Organization

This guide contains the following chapter:

- Chapter 1, “Using the Batch Refresh Process,” describes the Modernization Workbench Batch Refresh Process (BRP).

Conventions

This guide uses the following typographic conventions:

- **Bold type**: indicates a specific area within the graphical user interface, such as a button on a screen, a window name, or a command or function.
- *Italic type*: indicates a new term. Also indicates a document title. Occasionally, italic type is used for emphasis.
- `Monospace type`: indicates computer programming code.
- **Bold monospace type**: indicates input you type on the computer keyboard.
- **1A/1B, 2A/2B**: in task descriptions, indicates mutually exclusive steps; perform step A or step B, but not both.

Related Manuals

This document is part of a complete set of Modernization Workbench manuals. Together they provide all the information you need to get the most out of the system.

- *Getting Started* introduces the Modernization Workbench. This guide provides an overview of the workbench tools and discusses basic concepts. It also describes how to use common product features.
- *Preparing Projects* describes how to set up Modernization Workbench projects. This guide describes how to load applications in the repository and how to use reports and other tools to ensure that the entire application is available for analysis.

- *Analyzing Projects* describes how to analyze applications at the project level. This guide describes how to create diagrams of applications, how to perform change analysis across applications, and how to estimate application complexity and effort.
- *Analyzing Programs* describes how to analyze applications at the program level. This guide describes how to use HyperView tools to view programs interactively and perform program analysis in stages. It also describes how to set up an application glossary and how to extract business rules.
- *Managing Application Portfolios* describes how to build enterprise dashboards that track survey-based metrics for applications in your portfolio. It also describes how to use Enterprise View Express to browse Web-generated views of application repositories.
- *Creating Components* describes how to extract program components from a legacy application.
- *Transforming Applications* describes how to generate legacy application components in modern languages.
- *Error Messages* lists the error messages issued by Modernization Workbench, with a brief explanation of each and instructions on how to proceed.

Online Help

In addition to the manuals provided with the system, you can learn about the product using the integrated online help. All GUI-based tools include a standard Windows **Help** menu.

You can display:

- The entire help system, with table of contents, index, and search tool, by selecting **Help:Help Topics**.
- Help about a particular Modernization Workbench window by clicking the window and pressing the **F1** key.

Many Modernization Workbench tools have *guides* that you can use to get started quickly in the tool. The guides are help-like systems with hy-

perlinks that you can use to access functions otherwise available only in menus and other program controls.

To open the guide for a tool, choose **Guide** from the **View** menu. Use the table of contents in the **Page** drop-down to navigate quickly to a topic.

Using the Batch Refresh Process



The Modernization Workbench (MW) *Batch Refresh Process* (BRP) lets you register and verify source files in batch mode. You typically use this process when sources on the mainframe have changed, and you need to synchronize the modified sources with the sources you are working with in MW. You can also use BRP to perform analysis and reporting functions.

Understanding the Batch Refresh Process

The Batch Refresh Process is a utility that supports the synchronization of sources from a mainframe or enterprise server with the MW repositories representing those sources. It is installed with the MW Enterprise Edition Server. For background on the server, see *Getting Started* in the workbench documentation set.

BRP is responsible for updating the workspace with sources provided to it from the mainframe and verifying all unverified sources. Optionally, BRP can be configured to run any required source code pre-processing, as well as certain analysis and reporting functions. These options are detailed later in this chapter.

When sources are updated to a workspace, the workbench determines whether or not to load the file. When the name of the incoming file matches the name of a file currently in the workspace, the two files are compared. If they are different, the incoming file will replace the existing file. If they are the same, no change is made. If the incoming file does not currently have a match in the workspace, the file is added to the “default” project. The default project is a project with the same name as the workspace. If this project does not exist, it is automatically created.

Updating a source in a workspace causes that source to be invalidated or “unverified.” Any sources that are dependent upon the updated file will also be invalidated. For example, an update to a copybook will cause all the COBOL files that use the copybook to become invalidated. That, in turn, will cause all JCL files that execute the programs in the source files to become invalidated. Once the update phase is completed, all unverified sources in the workspace will be verified.

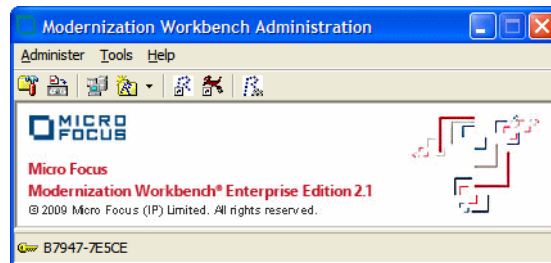
Configuring the Batch Refresh Process

The Batch Refresh Process is installed with the MW Enterprise Edition Server. For each workspace it processes, BRP refers to an INI file containing configuration settings. Use the BRP Configurator in the workbench Administration tool on the workbench server to modify the INI file settings.

To use the BRP Configurator:

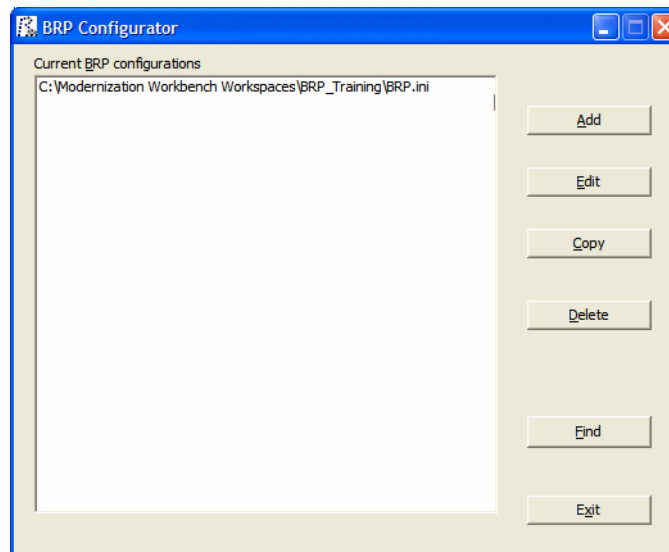
- 1 In the Windows **Start** menu, go to the **Programs** folder. Choose **Micro Focus**, then **Modernization Workbench Administration**. The Modernization Workbench Administration tool window opens (Figure 1-1).

Figure 1-1 *Administration Tool*



- 2 In the **Administer** menu, choose **Configure BRP**. The BRP Configurator opens (Figure 1-2).

Figure 1-2 *BRP Configurator*



- 3 In the Current BRP Configurations pane, choose the BRP configuration file you want to edit and click **Edit**. If the BRP configuration you want to edit is not listed in the pane, click **Find** to locate the file in the file system

1-4 Using the Batch Refresh Process
Configuring the Batch Refresh Process

Note: To create a new configuration, click **Add**. A Select Workspace dialog opens, where you can specify the workspace (.rwp file) you want to configure for BRP.

To copy a configuration, select it and click **Copy**. A Select Workspace dialog opens, where you can specify the workspace (.rwp file) you want to copy the configuration to. To delete a configuration, select it and click **Delete**.

- 4** The main BRP Configurator window opens, with a tab for each configuration task:
 - On the General tab, set basic BRP initialization values (required). See [“Configuring General Settings” on page 1-5](#).
 - On the User Exit tab, identify any user exits you have created to extend or modify BRP functionality (optional). See [“Configuring User Exits” on page 1-8](#).
 - On the Advanced tab, enable BRP support for IMS Analysis, Executive Report, and WebGen (optional). See [“Configuring Advanced Settings” on page 1-12](#).
- 5** When you are satisfied with your entries on each tab, click **OK**.

Configuring General Settings

Set required BRP initialization values on the General tab (Figure 1-3). The settings are described in Table 1-1.

Figure 1-3 *General Tab*

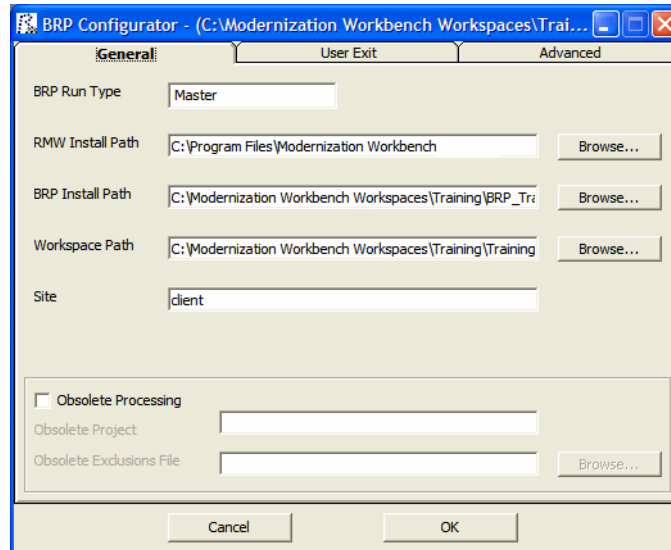


Table 1-1 *Required BRP Initialization Settings*

Setting	Description
BRP Run Type	Must be set to “Master”.
RMW Install Path	Specifies the MW install path.
BRP Install Path	Specifies the BRP install path. This directory must contain the Reports, PreparedSources, Staging, Utility, and Working folders. Use override parameters for those that are in a different location.
Workspace Path	Specifies the Workspace Repository path. This directory is in the same location as the workspace RWP file.
Site	Specifies the site name for this BRP install. This value is written out to the main BRP log and is used for documentation purposes only.

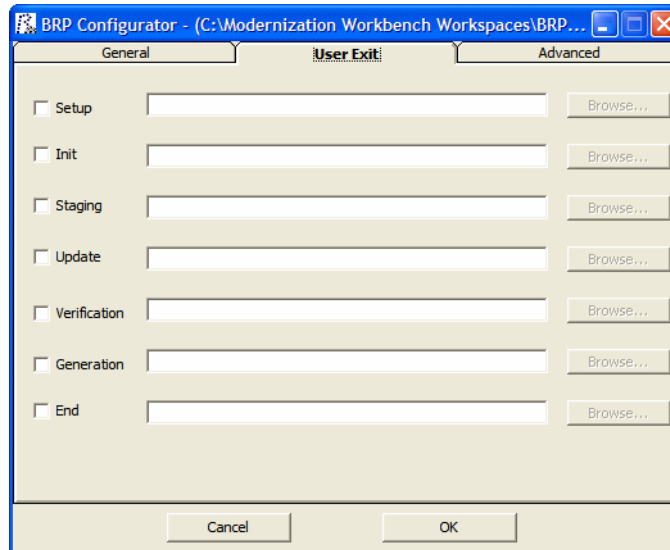
Table 1-1 Required BRP Initialization Settings

Setting	Description
Obsolete Processing	Check this box to turn on obsolete processing. Obsolete processing automatically determines which source files are no longer part of a “live” application and moves them to another project. Sources are determined to be obsolete by virtue of being absent from the set of incoming sources for a BRP run.
Obsolete Project	If Obsolete Processing is selected, specifies the project to which obsolete source files will be moved.
Obsolete Exclusions File	If Obsolete Processing is selected, specifies a text file that lists files that should be ignored during obsolete processing. This mechanism is intended to avoid having MW generated or provided files classified as obsolete. For example, MW-provided system copybooks or DASDL generated copybooks. The text file should be formatted with a single file name per line. This mechanism is also useful when there are sources that are particularly difficult to provide on an ongoing basis or if a source is generated during runtime in the application.

Configuring User Exits

Identify user exits you have created to extend or modify BRP functionality on the User Exit tab (Figure 1-4). A user exit is a point in the standard BRP processing when a user-supplied set of commands is executed. Typically the commands execute utilities that accomplish tasks ranging from source code pre-processing to specialized report generation.

Figure 1-4 *User Exit Tab*



Organization

There are seven user exits in BRP. Each is named and corresponds to a major division of processing, or step, in a BRP run. The names are listed below in the order they are executed:

- Setup
- Init (Initialization)
- Staging
- Update
- Verification
- Generation
- End

With the exception of the Setup and End user exits, each is executed as the very first task of the corresponding BRP step. For example, in the

Generation step the Generation user exit is executed followed by executive report generation and WebGen generation.

There are some essential tasks that occur during the Setup step that make it impractical for user exit execution to be first. The Setup step is where the main BRP log is opened and all parameter values are generated, if necessary, and checked for validity. The Setup user exit occurs *after* the log file is created, but *before* parameter values are generated and checked.

The End user exit occurs at the very end of the BRP run. There are no tasks that occur after it other than closing the main BRP log file.

Which user exit should be used to execute a particular piece of functionality depends upon the task that needs to be accomplished. For example, source code pre-processing usually needs to occur prior to the sources being loaded into the workspace. This would make the Staging or Update user exits ideal. However it is best to do source code pre-processing once all sources are in a single spot and are guaranteed to have proper file extensions. That would eliminate the Staging user exit, since it is during Staging that file extensions are added, if necessary. Therefore, the best place to execute source code pre-processing utilities is the Update user exit.

Other common uses of user exits are to run specific reporting or analysis functions. These typically require that the verification step has been completed. Therefore, the Generation user exit will typically work best for these situations.

Configuration

Configuring a user exit involves two separate tasks:

- Creating a BRP-enabled utility to accomplish the task at hand.
- Pointing the user exit to that utility.

Creating a BRP-enabled utility is a non-trivial task. Guidelines and information on this subject can be found in [“Producing Utilities for BRP” on page 1-18](#). Use the User Exit tab of the BRP Configurator to point a user exit to the corresponding BRP-enabled utility.

Note: In the default configuration the Generation user exit is configured and provides a useful example.

BRP contains anchor points for all seven user exits. The DOS batch file should be named for the user exit it corresponds to and it should be located in the BRP Utilities folder. Be sure to specify a full and complete path to the DOS file. Relative paths may not work properly in this context.

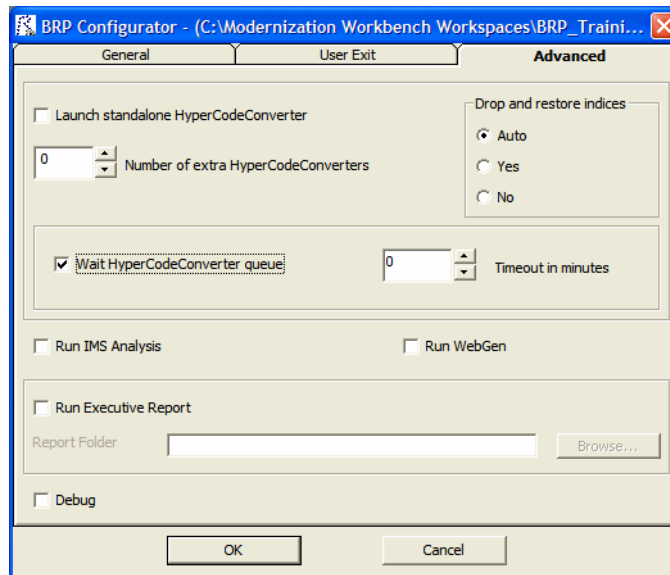
The DOS batch file must contain the actual commands that the user exit will execute. This also provides the opportunity to do more than one task in any given user exit.

BRP checks any enabled user exit INI file parameter value for validity during the Setup step. If the value does not point to an existing file, BRP will quit with a severe error.

Configuring Advanced Settings

Configure settings on the Advanced tab to improve verification performance and enable support for IMS Analysis, Executive Report, and WebGen (Figure 1-5). The settings are described in Table 1-2.

Figure 1-5 Advanced Tab



1-10 Using the Batch Refresh Process
Configuring the Batch Refresh Process

Table 1-2 *Advanced Settings*

Setting	Description
Launch standalone HyperCode Converter	Check this box to launch the HyperCode Converter. Using the HyperCode Converter generally improves verification performance.
Number of extra HyperCode Converters	If Launch standalone HyperCode Converter is selected, click the arrow buttons to specify the number of additional HyperCode Converters you want to launch.
Wait HyperCode Converter queue	Check this box to force BRP to wait until the HyperCode Converter(s) queue is empty.
Timeout in minutes	If Wait HyperCode Converter queue is selected, click the arrow buttons to specify the time in minutes BRP should wait for the count on the HyperCode Converter(s) queue to change. If the count does not change within the specified time, BRP resumes. Sixty minutes is recommended.
Drop and restore indices	<p>Dropping repository indexes generally improves verification performance when a large number of files need to be verified. Select:</p> <ul style="list-style-type: none"> • Auto if you want BRP to drop repository indexes based on the number of files that need to be verified. • Yes if you want BRP to drop repository indexes. • No if you do not want BRP to drop repository indexes. <p>Dropped indexes are restored when the verification process completes.</p>
Run IMS Analysis	Check this box to enable IMS Analysis.

Table 1-2 *Advanced Settings*

Setting	Description
Run WebGen	Check this box to enable WebGen.
Run Executive Report	Check this box to enable Executive Report.
Report Folder	Specifies the folder to store the Executive Report in. Use the Browse button to locate the folder.
Debug	Check this box to enable debug mode, in which additional messages are written to the BRP log.

Running the Batch Refresh Process

Running the Batch Refresh Process involves specific steps described in [“Preparation,”](#) [“Execution,”](#) and [“Execution.”](#)

Preparation

The incoming sources must be placed in the BRP PreparedSources folder with appropriate file extensions.

If files do not have appropriate file extensions, they must be separated, by type, into individual folders in the PreparedSources folder. Each subfolder must be named for the type of source it contains. Source names must match those used in the Sources folder in the workspace directory. For example, if Cobol sources do not have file extensions they must be placed in PreparedSources\Cobol.

Execution

The BRP Configurator creates a runBRP.bat file and saves it to the location specified in the BRPInstallPath (see Table 1-1). Executing this batch file will start a BRP run. This file executes the runBRP.exe executable with appropriate parameters. The command format is as follows:

```
runBRP.exe <INI file> <log file>
```

where *INI file* is the path to the INI file and *log file* is a path to where the main BRP log file is written.

A full BRP run will produce several detailed log files in addition to the main BRP log. These detail files will always be written to the Reports folder. The main BRP log also is written to the Reports folder by default.

Note: The workspace is locked while BRP runs. It cannot be accessed by users. In the event of BRP failure, you can unlock the workspace by choosing **Administer:Unlock Workspace** in the Administration tool.

Optional Processing

This section discusses the usage and configuration of the optional processing and reporting features of BRP.

Adding Source File Extensions

It is recommended that source files coming into BRP have proper file extensions already in place. In some cases, however, this is not possible and BRP can add them if needed. There is no need to configure INI file parameters to use the functionality.

To have BRP add the file extensions, you must separate the sources, by type, into separate folders in the PreparedSources directory. Each folder must be named for the source type it contains and the source type name must correspond to MW source type names.

MW source type names can be determined by examining the folder names found in the Sources folder of a workspace directory. If the workspace already contains a source of a particular type, there will be a folder in the Sources directory corresponding to that source type. For example, Cobol files are found in the Cobol folder. The precise file extension that is added for any particular source type is determined by the configuration of the Registration Extensions tab in the target workspace's workspace options. The first defined file extension for each source type will be the extension that is added by BRP. For example, Cobol File has three default file extensions listed: .cbl, .cob, and .ccp. Since .cbl is listed first, that is the extension used by BRP. The order that these values appear in

the workspace options can be changed by removing extensions and adding them back in.

Note that file extensions are added onto the file without regard for any currently existing file extension if this functionality is used. For example, if the files in a folder named Cobol currently have a .txt extension (which is commonly added by some mainframe FTP applications), each file would end up having an extension like .txt.cbl. Various source file naming conventions include multiple “dots” in the source name. Since this scenario is unpredictable and varies widely, it is risky and impractical to have BRP strip any possible existing file extensions.

If there is a mix of sources with and without file extensions, BRP can handle this. Any files with proper extensions should be placed in the PreparedSources directory directly, as normal. Any files that need extensions should be dealt with as described above.

BRP Logging

At the beginning of every BRP run a timestamp value is generated, consisting of the date and time. That timestamp is used throughout the run in order to uniquely identify and group the logs and information generated. Each run creates log files as well as other pieces of information. The timestamp for the run is added to the beginning of the filename for each log. All logs and information files are typically written to the Reports folder of the BRP install directory.

The main BRP log is, by default, named “BRP log.txt,” although the name can be altered by changing the command that is found in the run-BRP.bat file in the BRP install directory. If more than one BRP install is present it is recommended that the name of the application that is being refreshed be added to the main BRP log name.

The main BRP log contains basic information on when each major step of the run starts and finishes, as well as any relevant summary or diagnostic information. This is the log to check to determine whether the BRP run completed successfully or not. A run to completion will result in the last message in the log indicating the process finished successfully. Log messages marked “ERROR” should be reviewed. These are problems that were encountered, but they are not bad enough to cause BRP

to abend. Messages marked as “SEVERE” are issues encountered that required BRP to abend. These should be investigated and corrected.

In addition to the main BRP log, the “Update Log.txt” and “Verify Log.txt” are also generated. These, as their names indicate, document the results of the update and verification steps respectively. The update log contains an entry for each file that is added to the workspace, whether by virtue of being different (updated) or new (added). Files that are processed during update that have unknown file extensions will also be documented in this log. Files that are processed and rejected because they are not different from the version in the workspace are not documented. The verification log lists the verification status of each file that is processed during verification. Summary statistics appear at the end of the log.

The remainder of the logs and files that are generated during a BRP run are there to provide in-depth information for troubleshooting if there is a problem during the run. If there is a problem that requires the attention of support services, please be sure to include all the logs and files from a run. Sorting the file names in the Reports directory by name will naturally group them together.

Producing Utilities for BRP

The following are guidelines for producing utilities for BRP. These guidelines apply for any utility. Currently these utilities are normally written by support services and partners.

Versioning

Each utility needs a version number. The version number should be the date of the last modification made to the utility, formatted as follows:

yyyymmdd

The version number must appear in the first line of the log file that the utility produces.

Logging

Log files are often the only way to get reliable data. The task of analyzing output can become easier when the log files are used and recording appropriate levels of output.

At a minimum log files need to contain:

- Utility name
- Utility version
- Parameter names and values
- Record of files modified/written (when appropriate)
- Record of individual changes made to modified files (when appropriate)

The log message format should be as follows:

```
hh:mm:ss<tab>message type<tab>message
```

Message types can include INFO, WARN, ERROR, SEVERE or DEBUG. These are generally self-explanatory, but SEVERE should not be used unless there is an `abend` (in Perl the `die()` command). Add new message types if the situation calls for it. For example, BRP has a SETUP message type.

Source, Executable, and CFG Files

Utilities are produced by support services and partners and are delivered as a compiled executable with documentation and, if necessary, a CFG file.

Guidelines for BRP Utilities

This section will focus on guidelines for utilities that need to be “enabled” for BRP. The only difference is where input is coming from, output is going to, and how parameters are provided.

BRP and Non-BRP Modes

In general, any utility created for BRP should also be able to be run in a standalone manner; that is, it should run outside of and separate from BRP as well. Typically this means getting parameters from a CFG file. This is already being done for all pre-processing type utilities right now. There are occasional situations where this is not practical. The utility needs to be able to determine whether it is being executed in a BRP context or not. If the stand-alone mode requires a CFG file, the absence of a CFG file parameter can serve as a trigger for BRP-mode execution. Where this will not work, the first parameter of the utility should be “BRP” to trigger BRP-mode execution.

Using User Exits

There are several user exit points in BRP; see [“Configuring User Exits.”](#) At different user exits potential input files are in different places and output requirements are different as well. Knowing which user exit a utility is going to be run from is crucial. It is recommended that support services be consulted regarding which user exit to employ for a particular task. The majority of user exit utilities are source code pre-processors and all use the “Update” user exit.

Parameter Data

Parameter data typically comes from any of three general sources: command line, CFG file, or DOS environment variables. The first two are straightforward. DOS environment variables are easily acquired by capturing the output of the DOS set command with the following line of Perl code:

```
$dos_env_vars_str = `set`;
```

Note: The special characters preceding and following the word “set” are not single quote characters; they are “backtick” characters.

The parameter values that drive a BRP run are made available to a user exit via DOS environment variables. BRP generates a DOS batch file that contains commands to set DOS environment variables. The user exit command is added to the end of the generated batch file and the batch

file is executed using the backtick operator in Perl. The backtick operator executes a DOS command (in BRP the path to a batch file) in a shell “nested” inside of the shell of the BRP executable. The environment variables set up for a user exit only exist during the execution of that user exit. The environment variable commands are re-generated and run for each user exit.

In general, the format of parameter names and values should be standardized. BRP job parameters are of the form:

Parameter Name = Parameter Value

DOS environment variables and CFG file parameters are formatted in the same way. Command line parameters should follow the same standard. In general, command line parameters need to override the same named parameter from a CFG file or DOS environment. This allows a way to alter behavior in cases where the user may not have direct control over all the values.

Logging

In addition to the general logging guidelines the name of the log file and where it is written need to be addressed in BRP mode utilities.

The log file name pattern is: *(timestamp)UtilityName* Log.txt

where *UtilityName* is obvious and *timestamp* is a BRP provided value (BRP_TIMESTAMP) that identifies all logs for a BRP run. See the section below on [“BRP Environment Parameters”](#) for details.

Input/Output

Input and output locations will change depending on what files are needed and which user exit the utility is run from. Most utilities (source code pre-processing) will be running from the Update user exit.

The sources coming into the BRP process will be in the following path:

BRP_STAGINGDIR\BRP_TIMESTAMP

where *BRP_STAGINGDIR* is a full path referring to the Staging folder of a BRP install and *BRP_TIMESTAMP* holds the timestamp value for the current BRP run.

Output sources must be written back to this same location. However, to maintain integrity should the user-exit utility fail or otherwise not finish, it is recommended that output sources be written to the BRP Working folder (BRP_WORKINGDIR) and only when processing is completed should they then be copied back to the proper output location. A subfolder should be created in the Working folder for this purpose using the following format:

timestamp_UTILITYName

This naming convention is required.

Returning Values

BRP determines the return state of a user exit by examining all the output written to the “console” (STDOUT in Perl terminology) by the commands executed by the user exit. The examination is done after the user exit completes execution and control returns back to the BRP run. If there is no output BRP assumes the user exit commands completed successfully. If there is any output found BRP assumes there was a SEVERE level error and will immediately stop the run.

BRP will include any output it finds in a SEVERE level message in the main BRP log. Any user exit executed utility should be sure to make effective use of this behavior. User exits do not have any knowledge of what commands or utilities they are executing. Therefore a message written to the console should contain the utility or command name along with an appropriately brief message. The details behind a utility failure can be included in the utility’s own log.

BRP Environment Parameters

BRP parameter values are split into two groups. Ones prefixed with “BRP_” are for BRP specific values. Those prefixed with “EXT_” are for source file extension definitions.

Table 1-3 lists all variables that are set by BRP for use by user exits along with a short description. Any path value will be fully qualified unless otherwise noted:

Table 1-3 *BRP Environment Variables*

Category	Name	Description
Timestamp	BRP_TIMESTAMP	Timestamp value that uniquely identifies a BRP run and the logs that are generated during that run.
BRP Logs		These parameters are the paths to log files from running various BRP. Note that all the references to specific jobs are default settings only. There are very few cases where these exact jobs will not be used, but they do exist.
	BRP_APPLYOBSOLETEPCFLOGFILE	Log from ApplyPCF.bj job for applying the BRP_OBSOLETEPCF file.
	BRP_BWGLOGFILE	Log from BWG.exe (Batch Web-Gen).
	BRP_CREATEBEGINPCFLOGFILE	Log from CreatePCF.bj job. This is run at the beginning of the BRP run and creates the file BRP_BEGINPCF.
	BRP_CREATEENDPCFLOGFILE	Log from CreatePCF.bj job. This is run just before verification and creates the file BRP_ENDPCF.
	BRP_EXECREPORTLOGFILE	Log for the ExecutiveReport.bj job.

1-20 Using the Batch Refresh Process
Guidelines for BRP Utilities

Table 1-3 *BRP Environment Variables*

Category	Name	Description
	BRP_GETEXTLOGFILE	Log for the GetExtensions2.mbu job. This is run at the beginning of the BRP and creates the file BRP_FILEEXTFILE. See the section below on extension values for more information.
	BRP_IMSANALYSISLOGFILE	Log for the “IMS Analysis.bj” job.
	BRP_UPDATELOGFILE	Log for the UpdateOnly.bj job.
	BRP_VERIFYLOGFILE	Log for the VerifyOnly2.bj job.
PCF files		These are parameters for the various PCF files that are generated and used during a BRP run.
	BRP_BEGINPCF	Generated at the beginning of a BRP run. Used for many purposes in BRP including determining obsolete sources.
	BRP_ENDPCF	Generated toward the end of a BRP run., after the BRP_OBSOLETEPCF is applied and before verification.
	BRP_OBSOLETEPCF	Generated during a Master BRP run if BRP_OBSOLETEPROCESSING is set to 1. This will shift sources missing from the current incoming set of files to the project specified in BRP_OBSOLETEPROJECT.

Table 1-3 *BRP Environment Variables*

Category	Name	Description
BRP Files, Folders, Flags, and Log		These parameters are data files that BRP uses, flags that turn certain processing on or off, BRP install folders and other various values.
	BRP_DROPIND	Flag (1/0) that drops database indexes to improve verification and IMS Analysis performance.
	BRP_LAUNCHHCC	Flag (1/0) that launches the HyperCode Converter to improve verification performance.
	BRP_WAITHCC	Time in minutes to wait for HyperCode Converter to respond.
	BRP_BRPLOGFILE	Main log file for a BRP run.
	BRP_BRPINSTALLPATH	Path where BRP is installed. This is specified in the BRP INI file. BRP will derive the path values for the six BRP folders (Prepared-Sources, Reports, Staging, Utilities and Working) based off this path if they are not specified in the INI file.
	BRP_PREPAREDSOURCESDIR	Path where sources coming into BRP start off. In cases where a utility is in place to handle getting the sources off a server or main-frame this is the location where those sources are copied to.

1-22 Using the Batch Refresh Process
Guidelines for BRP Utilities

Table 1-3 *BRP Environment Variables*

Category	Name	Description
	BRP_REPORTDIR	Path to the folder where all log files are written to. Other resource files created during a BRP run are also written here including all PCF files, file extension data file and all generated user exit batch files.
	BRP_RMWINSTALLPATH	Path to the install folder for MW.
	BRP_STAGINGDIR	Folder where sources reside for updating to the workspace. Sources will actually be in a subfolder that is named with the BRP timestamp and not the Staging folder directly. This is also where the majority of source pre-processing utilities will look for inputs and write outputs.
	BRP_UTILITIESDIR	Path to directory that contains all the executables the BRP will need along with extra resource files and the static user exit batch files.
	BRP_WORKINGDIR	Folder \user exit utilities should use for any work they need to perform.
	BRP_WORKSPACEDIR	Path to the target workspace.
	BRP_FILEEXTFILE	File containing the file extension definitions for the target workspace.
	BRP_LASTRUNFILE	Text file containing the timestamp of the last BRP run that completed execution.

Table 1-3 *BRP Environment Variables*

Category	Name	Description
	BRP_OBSOLETEEXCLUSIONSFILE	File that lists any files that should be excluded from obsolete processing. Typically this includes files generated by MW (ex. DASDL copybooks), but is often used for client-specific sources as well.
	BRP_BRPRUNTYPE	Must be set to “Master”.
	BRP_DEBUG	Flag (1/0) that will increase the amount of messaging written to the main BRP log. Typically this is always set to 1.
	BRP_LASTRUNTIMESTAMP	The timestamp value of the last BRP run that completed execution.
	BRP_OBSOLETEPROCESSING	Flag (1/0) that turns obsolete processing on or off. When it is turned on the incoming set of files will be compared against the set of file currently in the target workspace. Any files currently in the workspace, but not in the incoming set of files will be moved to an obsolete project (named in the BRP_OBSOLETEPROJECT parameter).
	BRP_OBSOLETEPROJECT	Name of a project where obsolete sources will be moved to.

1-24 Using the Batch Refresh Process
Guidelines for BRP Utilities

Table 1-3 *BRP Environment Variables*

Category	Name	Description
	BRP_SITE	Documentation parameter that is set in the BRP INI file. The value here will be written to the beginning of the main BRP log file. It is used mainly for support purposes to ensure that INI and log files produced by a client match up.
User Exits		These parameters contain the command that will be executed by BRP. Typically this will specify a static DOS batch file (as opposed to the generated batch file BRP generates for each user exit). The static batch file is used so that multiple commands can be executed in a single user exit. These parameters will only exist if the user exit is being used. Below is a complete list of all user exit parameters, but it will be rare to see them all at once. They are listed in the order they would be executed in a BRP run. There is one user exit for each major step of the BRP. They always are the first task that is done in each step.
	BRP_USEREXIT_SETUP	
	BRP_USEREXIT_INIT	
	BRP_USEREXIT_STAGING	
	BRP_USEREXIT_UPDATE	
	BRP_USEREXIT_VERIFICATION	

Table 1-3 *BRP Environment Variables*

Category	Name	Description
	BRP_USEREXIT_GENERATION	
	BRP_USEREXIT_END	
File Extensions		These parameters contain information on the source file extensions that are valid for each legacy file type for the target workspace. The exact parameters that will be here depend upon what options are activated for the target workspace.
	Parameter Names	The general format of the parameter names is EXT_ <i>type</i> , where <i>type</i> is the name of the corresponding directory in the workspace Sources folder. Note that in the past this name is not necessarily the same as the type name found in a PCF file. For example, PL/I-included sources are contained in the Sources folder PLIInclude, but that source type is named “PLINC” in PCF files.
	Parameter Values	The extension values are separated by a single space and will be in the same order they appear in the workspace options window. Example: EXT_COBOL = cbl cob ccp C74

Testing

Testing user exit utilities can be challenging. The easiest way to do this is to use one of the DOS batch files generated for each enabled user exit during a BRP run. These files will contain all the parameters and the val-

1-26 Using the Batch Refresh Process
Guidelines for BRP Utilities

ues can be changed to suit the needs of the testing requirements. Replace the last command in the file with whatever command is necessary. In a Perl context we would use:

```
perl -d myUtility.pl
```

Note that a DOS command window will not execute one of these generated batch file when they have same filename the BRP run assigns (i.e. (timestamp)UserExit.bat). This is due to some intrinsic interpretation of the leading “(timestamp)” in the file name. Simply delete this portion of the file name and the batch file will work normally.

1-28 Using the Batch Refresh Process
Guidelines for BRP Utilities

1-30 Using the Batch Refresh Process
Guidelines for BRP Utilities

Index

B

Batch Refresh Process
 configuring **1-2**
 log files **1-13**
 overview **1-1**
 running **1-11**
 user exits **1-6**
 utilities **1-14, 1-15**
BRP Configurator **1-2**

C

configuration **1-2**

L

log files **1-13**

U

user exits **1-6**
utilities **1-14, 1-15**

Index-2