



Why monitor utility syntax?

Enforce and Maintain Company-Wide DB2 Utility Standards.

Jennifer Nelson
Product Specialist, Rocket Software
© 2010 IBM Corporation
© 2010 Rocket Software, Inc.

CONTENTS

List of Figures.....	iii
List of Examples.....	iv
1 Why Monitor DB2 Utility Syntax?.....	1
1.1 Automatically Block and Cancel Threads	1
1.2 Extend syntax for the LOAD, REORG and CHECK DATA utilities.....	1
1.3 Monitor utility syntax for compliance to company standards	2
2 What does the Utility Monitor do?.....	2
2.1 How the Utility Monitor Works.....	2
2.1.1 The Policy Rules are Used to Invoke the Utility Monitor	3
3 How to read and understand the Policy.....	3
3.1 Syntax Diagram Option Descriptions.....	8
3.2 Sample UET Policy to Invoke the Utility Monitor	11
4 How to Maximize the Benefits of the Utility Monitor	14
4.1 Example Practice Rules For Use with the Utility Monitor.....	14
4.2 Journaling Utility Monitor Actions.....	18
4.3 How Whitespace is Compressed From Utility Syntax.....	19
5 Appendix A: Sample Policy	vi
5.1 Hints and Tips about Policy Rules	xiii

LIST OF FIGURES

Figure 1: Substituting one text string for another.....	14
Figure 2: Conditionally failing a utility by syntax parameter.....	15
Figure 3: Conditionally failing a utility by user ID.....	15
Figure 4: Unconditionally failing a utility.....	16
Figure 5: Unconditionally failing a utility based on utility command.....	16
Figure 7: Removing a text string from the utility syntax.....	17
Figure 8: Adding a text string that is not present.....	17
Figure 9: Ensuring a text string is either changed or added.....	17
Figure 10: Default journaling at the utility level.....	18
Figure 11: Journaling at the utility level.....	18
Figure 12: Journaling at the utility level.....	18
Figure 13: Journaling at the utility level with an override at the syntax level.....	19
Figure 14: Sample SYSIN in DSNUTILB JCL.....	19
Figure 15: What the SYSIN looks like after it is normalized.....	19
Figure 16: Policy rule that will match normalized SYSIN.....	19
Figure 17: Policy rule that will not match normalized SYSIN.....	19

LIST OF EXAMPLES

Example 1: Sample portion of a Policy for use with the Utility Monitor.....	11
Example 2: Sample Policy used by the DSNUTILB Intercept.	vi

1 Why Monitor DB2 Utility Syntax?

Companies today are forced with many challenges when it comes to managing their complex DB2 for z/OS environments. Most DB2 applications are global, requiring almost 100% accessibility. Those demands reduce the amount of time available to perform necessary routine tasks, such as utility maintenance on the underlying data and objects stored in DB2 for z/OS that support critical business applications.

The DB2 Utilities Enhancement Tool for z/OS provides the Utility Monitor to address the needs of companies who wish to streamline their utility processing. The Utility Monitor allows you to establish and maintain company-wide DB2 utility syntax policies. It provides senior-level DB2 professions the confidence to know that utility jobs can be set and executed by junior-level staff, reducing effort and errors. It also provides a mechanism to assist with auditing DB2 utility processes.

In addition to the Utility Monitor, the DB2 Utilities Enhancement Tool enables you to manage your maintenance window and utility processing by canceling active threads on your objects, and blocking additional threads from coming active. It also extends the functionality of the LOAD, REORG and CHECK DATA utilities.

1.1 Automatically Block and Cancel Threads

Using rules defined in a utility intercept Policy, you can specify how you want UET to block and cancel threads. You can define the rules by object name, object type, user ID, etc. By using the many ways to define rules to the Policy to block and cancel threads, you can take control of your maintenance window and ensure that you have access to your business-critical data.

Not only does the DB2 Utilities Enhancement Tool cancel and block threads for DB2 objects, it extends the native utility syntax of the LOAD, REORG and CHECK DATA utilities.

1.2 Extend syntax for the LOAD, REORG and CHECK DATA utilities

In addition to automatically canceling active threads and blocking additional threads from becoming active, the DB2 Utilities Enhancement Tool has extended the capabilities of the LOAD utility, the REORG utility and the CHECK DATA utility in the following manner.

UET extends the native DB2 LOAD syntax by providing the following additional parameters:

- The CONSTANT and VALUEIF parameters replace the value for a particular field in the input records with another value that you specify.
- The PRESORT parameter sorts rows by table object identifier (OBID) and by clustering index key (or if no clustering index key is available, by the oldest defined index). The presorting of data can help reduce overall elapsed time of the LOAD utility.

UET enhances the DB2 REORG utility by providing the following function:

- Automatically sizes and creates the mapping table and mapping-table index that are required for the DB2 REORG TABLESPACE utility when the SHRLEVEL CHANGE option is specified. Also automatically drops these objects when reorganization processing completes to conserve space.

UET extends the native DB2 CHECK DATA syntax by providing the following additional parameters:

- DISCARDTO and DISCARDSpace parameters cause data rows in violation of check constraints to be written to an UNLOAD-format file instead of a DB2 table. The file can then be processed by the LOAD utility or another user-defined process.

1.3 Monitor utility syntax for compliance to company standards

New with APAR PM09829, you can now ensure your DB2 utilities are running with the proper utility syntax by activating the new Utility Monitor. With the Utility Monitor, DBAs can control not only what utilities get run, but who can run DB2 utilities and the parameters with which they must be run.

Using the DSNUTILB Intercept Policy, rules can be defined to specify which utilities should be monitored, on which objects, using what syntax parameters and by whom for an added layer of security to ensure good shop standards are being practiced and company policies are enforced.

2 What does the Utility Monitor do?

Each company's DB2 department enforces standards to protect business processes required to conduct business critical to the company's bottom line. These standards might include monitoring a user's actions against certain database objects, limiting user access to specific data or controlling the kinds of DB2 utilities that can be run against objects pertinent to daily business activity.

The Utility Monitor with the DB2 Utilities Enhancement Tool can assist DBAs and IT management by enforcing company standards and policies in regards to how the DB2 utility DSNUTILB is run. By defining rules within the intercept Policy, DBAs can control the authority to execute DB2 utilities as well as enforce which parameters are used with specific DB2 utilities or objects. DBAs can also opt to purposely fail a DB2 utility if it violates the rules defined within the Policy.

For example, DBAs can create a company standard that no REORG utility should run without the KEEPDICTIONARY parameter, or that no LOAD utility should run with the LOG YES parameter. Creating rules within the Policy such as these ensures that each DB2 utility that is run matches the definition of the Policy. And if they don't, Utility Monitor can either change the utility syntax to match the company standards, or fail the utility.

2.1 How the Utility Monitor Works

The DB2 Utilities Enhancement Tool has a Policy that governs the behavior of the UET DSNUTILB Intercept component. The Policy is a set of rules written in a language called XML that defines how UET intercepts a DSNUTILB utility and what actions it is to perform. The rules that are defined within the Policy dictate which utilities are monitored as well as what syntax to enforce. When a DSNUTILB utility job is run, UET evaluates the utility statement syntax along with the Utility Monitor rules within the intercept Policy. Each Rule within the Policy relates some utility statement text with a coded option. When the Rule is evaluated as true, the option directs UET to perform one or a combination of the following actions:

ADD

A text string is added to the utility statement if the specified string is not found in the utility statement syntax. The text string is appended to the end of the utility statement only once. For example, if a REORG utility does not have the parameter

KEEPDICTIONARY specified, UET will add the syntax KEEPDICTIONARY, and pass it to the REORG utility for processing.

REMOVE

A text string is removed from the utility statement if the specified string is found in the statement syntax. For example, if the COPY utility contains the parameter SHRLEVEL CHANGE, the parameter can be removed, causing the default value of SHRLEVEL REFERENCE to be used instead.

SUBSTITUTE

A text string is replaced in the utility statement if the specified string is found in the statement syntax. The substituted text is placed in the same location within the utility statement as the target string. For example, if a LOAD utility has the LOG YES parameter specified, UET will substitute LOG NO in place of LOG YES and pass it to the LOAD utility for processing.

JOURNAL

Indicates to UET to write this event to the UET DB2 table for subsequent review or analysis. The information written to the UET table will include pertinent information about the utility statement and environment, plus the original syntax specified and substituted syntax that was passed to the IBM utility and processed.

FAIL

Causes the utility job step to fail, and the return code value supplied in this element will be issued along with messages containing information about why the utility failed.

2.1.1 The Policy Rules are Used to Invoke the Utility Monitor

When the DB2 Utilities Enhancement Tool is installed and customized, a Policy is created in a language called XML containing default rules is created in the SABPSAMP library. The Policy is used by the started task when the DSNUTILB Intercept is invoked. The Policy rules dictate how UET intercepts a DSNUTILB utility and the actions it performs. The default Policy as shipped does not cause threads to be blocked and canceled, nor does it invoke the Utility Monitor. It must be customized before any action is taken by the UET started task.

By default, the name of the Policy is created as *abpidPLCY*, where *abpid* is the 4-character identification for your installation of UET. For example, an ABPID could be named ABP1. In this scenario, the Policy name would be ABP1PLCY.

If any of the rules within the Policy evaluate as true, the action described by the Policy Rule will invoke the Utility Monitor.

3 How to read and understand the Policy

The DSNUTILB Intercept Policy is what is used by the UET started task to understand which types of threads to cancel and block for each DB2 subsystem, as well as what utilities to monitor. The Policy is written in XML which is a language that tags elements of a

document that then allows a parser to read and understand the instructions in the document. In the Policy, there are different kinds of elements with various values that do different things within UET. When those elements are grouped together, UET interprets those elements as Rules.

Within an XML document such as the UET Policy, there are elements, attributes, values, and hierarchical relationships between elements. While XML is beyond the scope of this whitepaper, definitions of each of the XML components used within UET's Policy are defined here for comprehensiveness.

ELEMENT

An element is a tag, along with its attributes and the text and elements that it encloses, that defines a characteristic of a document. Elements within another element are called child elements.

CHILD ELEMENTS

Child elements are tags that define characteristics of a document, but are only specified when the parent element is defined. That is, they cannot be specified on their own, without the parent element being declared first.

ATTRIBUTE

An attribute is the name-value pair that immediately follows an element name. Attributes enhance the specification of an element.

VALUE

An attribute has values, or valid options that may be used within an attribute, much like a parameter has options.

The XML elements below help explain the Policy used by UET. A sample UET Policy is also discussed in Appendix A.

Element DB2SYSTEM

Used by element	POLICY
Child elements	VRUPDATE, USE PRACTICE, USE RULESET, INCLUDE, EXCLUDE
Attribute 1 Required/Optional Value	SSID Required SSID supplied by the user in quotes
Attribute 2 Required/Optional Values	ACTION Optional BLOCK_AND_CANCEL_THREAD (default) VRUPDATE MONITOR_UTILITY

Element DSNUTILB_INTERCEPT

Used by element	None; this is the root element, or the top-level element in the Policy.
Child elements	RULESET, PRACTICE, POLICY
Attribute	None

Element EXCLUDE

Used by elements	DB2SYSTEM, RULESET
Child element	RULE
Attribute	None

Element INCLUDE

Used by element	DB2SYSTEM, RULESET
Child element	RULE
Attribute	None

Element MONITOR

Used by element	UTILITY
Child element	SYNTAX
Attribute 1 Required/Optional Values	JOURNAL Optional YES (default), NO
Attribute 2 Required/Optional Value	FAIL Optional Value between the range of 8 and 4095

Element POLICY

Used by element	DSNUTILB_INTERCEPT
Child element	DB2SYSTEM
Attribute	None

Element PRACTICE

Used by element	DSNUTILB_INTERCEPT
Child element	UTILITY
Attribute Required/Optional Value	NAME Required Practice Name supplied by the user in quotes. Length 32 characters.

Element RULE*

Used by elements	EXCLUDE, INCLUDE
Child element	None
Attribute 1 Required/Optional Value	UTILITY_JOBNAME Optional Jobname supplied by the user in quotes
Attribute 2 Required/Optional Value	UTILITY_USERID Optional User ID supplied by the user in quotes
Attribute 3 Required/Optional	UTILITY_ID Optional

Value	DB2 utility ID supplied by the user in quotes
Attribute 4 Required/Optional Values	SHRLEVEL Optional ALL (default), NONE, REFERENCE, CHANGE
Attribute 5 Required/Optional Values	UTILITY_COMMAND Optional ALL (default), CHECK_DATA, CHECK_INDEX, CHECK_LOB, COPY, EXEC_SQL_ALTER, EXEC_SQL_DROP, LOAD, QUIESCE, REBUILD_INDEX, RECOVER, REORG_INDEX, REORG_INDEXSPACE, REORG_TABLESPACE, UNLOAD
Attribute 6 Required/Optional Value	DATABASE Optional Database name supplied by the user in quotes
Attribute 7 Required/Optional Value	TABLESPACE Optional Tablespace name supplied by the user in quotes
Attribute 8 Required/Optional Value	INDEXSPACE Optional Indexspace name supplied by the user in quotes
Attribute 9 Required/Optional Value	INDEX Optional Index name supplied by the user in quotes
Attribute 10 Required/Optional Value	TABLE Optional Table name supplied by the user in quotes in the format <creator>.<table> or <table>.
Attribute 11 Required/Optional Value	VIEW Optional View supplied by the user in quotes
Attribute 12 Required/Optional Value	ALIAS Optional Alias supplied by the user in quotes
Attribute 13 Required/Optional Value	SYNONYM Optional Synonym supplied by the user in quotes
Attribute 14 Required/Optional Value	PART Optional Part number supplied by the user in quotes.. For more information on how PART may be specified within a RULE, please refer to the manual entitled <i>IBM DB2 Utilities Enhancement Tool User's Guide</i> , document number SC19-1224, in the chapter regarding the DSNUTILB Intercept Policy.

* At least one Attribute is required if the element RULE is specified.

Element RULESET

Used by element	DSNUTILB_INTERCEPT
Child elements	EXCLUDE, INCLUDE
Attribute Required/Optional Value	NAME Required Ruleset Name provided by user in quotes

Element SYNTAX*

Used by element	MONITOR
Child element	None
Attribute 1 Required/Optional Value	ADD Optional String to add supplied by the user in quotes
Attribute 2 Required/Optional Value	REMOVE Optional String supplied by the user in quotes
Attribute 3 Required/Optional Value	VALUE Optional Comparative value supplied by the user in quotes
Attribute 4 Required/Optional Value	SUBSTITUTE Optional Substituted value supplied by the user in quotes
Attribute 5 Required/Optional Values	JOURNAL Optional YES (default), NO
Attribute 6 Required/Optional Value	FAIL Optional Numeric value supplied by the user in quotes. Value between the range of 8 and 4095.

* At least one Attribute is required if the element SYNTAX is specified.

Element USE_PRACTICE

Used by element	DB2SYSTEM
Child element	None
Attribute 1 Required/Optional Value	NAME Required Practice Name supplied by the user in quotes

Element USE_RULESET

Used by element	DB2SYSTEM
Child elements	None
Attribute Required/Optional Value	NAME Required Ruleset Name supplied by the user in quotes

Element UTILITY

Used by element	PRACTICE
Child element	MONITOR
Attribute Required/Optional Value	NAME Required CHECK_DATA, CHECK_INDEX, CHECK_LOB, COPY, EXEC_SQL_ALTER, EXEC_SQL_DROP, LOAD, QUIESCE, REBUILD_INDEX, RECOVER, REORG_INDEX, REORG_INDEXSPACE, REORG_TABLESPACE, UNLOAD

3.1 Syntax Diagram Option Descriptions

The following options describe the three types of document tags used to create the Policy Rules: elements (and child elements), attributes and values.

DB2SYSTEM SSID="ssid"

The element DB2SYSTEM specifies the DB2 SSID on which to perform the DSNUTILB Intercept processing to block and cancel threads or to monitor DB2 utilities. This element is required. There is no default value.

ACTION BLOCK AND CANCEL THREADS | MONITOR_UTILITY

The element ACTION is a child element to DB2SYSTEM. It specifies the action to perform on the specified subsystem. This element is optional, however, if it is not specified, the default value is BLOCK_AND_CANCEL_THREADS.

BLOCK_AND_CANCEL_THREADS

The attribute BLOCK_AND_CANCEL_THREADS is the default value and specifies that threads should be blocked and canceled based on the criterion defined in the elements RULE and/or RULESET.

MONITOR_UTILITY

The attribute MONITOR_UTILITY specifies the utility or utilities on which syntax should be monitored, and is invoked when the criterion in the RULE and/or RULESET matches the utility being executed.

DSNUTILB_INTERCEPT

The element DSNUTILB_INTERCEPT appears at both the beginning and the end of the Policy Rule definitions, and indicates that the Policy rules defined between the elements are to be included as part of the Policy. In XML terms, this is what is called the "root element".

MONITOR JOURNAL | FAIL | SYNTAX

The element MONITOR specifies what parameters to monitor within the element UTILITY NAME. This element is required when invoking the Utility Monitor and can only be specified once per UTILITY element. There is no default value.

JOURNAL YES | NO

This attribute of MONITOR specifies that logging is to take place to track the actions of the Utility Monitor activities. This element is optional. The default value is YES. If JOURNAL is set to YES, then a row will be written to the UET Journal tables. If JOURNAL is specified with the MONITOR element, it will take precedence for all SYNTAX elements contained within the MONITOR element. If JOURNAL is specified on the SYNTAX element, it will override any value specified in the MONITOR element for that syntax activity only. This attribute is only specified once with either element MONITOR or SYNTAX.

FAIL

This attribute of MONITOR specifies to fail the job step when passing control to DSNUTILB. Valid values are an integer in the range of 8 to 4095. For example,

FAIL="20". The utility job step return code will be set to the value defined by the attribute FAIL. If FAIL is specified for the MONITOR element, the value becomes the default value for all subsequent SYNTAX elements but can be overridden if it is also specified in the child element SYNTAX. This attribute is optional. There is no default value. This attribute is only specified once with either element MONITOR or SYNTAX.

SYNTAX ADD | REMOVE | VALUE | JOURNAL

This child element of MONITOR specifies what syntax to monitor within the specified UTILITY NAME. Valid attributes are ADD, REMOVE or VALUE concatenated with attribute SUBSTITUTE. This element is optional. There is no default value. The element SYNTAX may be specified multiple times within the element MONITOR. If the element Syntax is specified, at least one of the following attributes is required.

ADD

This attribute of element SYNTAX specifies the text string to add to the utility syntax being executed by DSNUTILB if the text string is not present. If the text string already exists, no action is taken. This attribute is optional, but if the element SYNTAX is specified, then ADD, REMOVE or VALUE must also be specified. There is no default value. This attribute may only be specified once per SYNTAX element.

REMOVE

This attribute of element SYNTAX specifies the text string to remove from the utility syntax being executed by DSNUTILB if the text string is present. If the text string is not present in the utility syntax, no action is taken. This attribute is optional, but if the element SYNTAX is specified, then ADD, REMOVE or VALUE must also be specified. There is no default value. This attribute may only be specified once per SYNTAX element.

VALUE SUBSTITUTE | FAIL

This attribute of element SYNTAX specifies the text string on which to match within the utility syntax before one of two actions takes place: substitute the target value for another one, or fail the utility. This attribute is optional, but if the element SYNTAX is specified, then ADD, REMOVE or VALUE must also be specified. There is no default value. This attribute may only be specified once per SYNTAX element.

If the attribute VALUE is specified but the attributes SUBSTITUTE or FAIL are not specified, the default attribute is JOURNAL.

SUBSTITUTE

This attribute of element SYNTAX specifies the text string to substitute in the utility if there is a match in the string provided for the attribute VALUE. This attribute is optional, but either SUBSTITUTE or FAIL must be supplied if VALUE is defined. There is no default value.

FAIL

This attribute of element SYNTAX specifies to fail the job step when passing control to DSNUTILB. Valid values are an integer from 8 to 4095. For example, FAIL="20". The utility job step return code will be set to the value defined by the attribute FAIL.

This specification of FAIL will override any value specified in the MONITOR element for this syntax activity only. This attribute is optional,

but either SUBSTITUTE or FAIL must be supplied if VALUE is defined. There is no default value.

JOURNAL YES | NO

This attribute of element SYNTAX specifies that logging is to take place to track the actions of the Utility Monitor activities. This attribute is optional. The default value is YES. If JOURNAL is set to YES, then a row will be written to the UET Journal tables. If JOURNAL is set to NO, then no logging activity will occur.

If JOURNAL is specified with the element MONITOR, it will take precedence for all SYNTAX elements contained within the MONITOR element. If JOURNAL is specified on the element SYNTAX, it will override any value specified in the MONITOR element for that syntax activity only. This attribute may only be specified once per SYNTAX element.

POLICY

The element POLICY appears both before and after the DB2 subsystem-specific rules which identifies the rules that are to be evaluated for DSNUTILB interception. This element may only be defined once within the element DSNUTILB_INTERCEPT.

PRACTICE NAME="name"

The element PRACTICE specifies the name of the Rule that contains subsequent syntax actions. This element is required when invoking the Utility Monitor. There is no default value. This element may be defined multiple times within the element DSNUTILB_INTERCEPT. This element may be defined only once within the element DB2SYSTEM.

UTILITY NAME="utility"

This child element of PRACTICE specifies the DB2 utility that UET is to evaluate when DSNUTILB is invoked to verify if the Practice Rule should be enforced. This element is required to invoke the Utility Monitor. This element may be defined multiple times within the element PRACTICE. There is no default value. Valid values are CHECK_DATA, CHECK_INDEX, CHECK_LOB, COPY, EXEC_SQL_ALTER, EXEC_SQL_DROP, LOAD, QUIESCE, REBUILD_INDEX, RECOVER, REORG_TABLESPACE, REORG_INDEX, REORG_INDEXSPACE, and UNLOAD.

RULE *object_type*="value"

The element RULE defines the type of object on which to cancel and block threads, or invoke the Utility Monitor. The object type can be an alias, a database, a table or tablespace, an indexspace, the SHRLEVEL value, or any one of the valid types of objects listed in Table 10 in the IBM manual entitled *IBM DB2 Utilities Enhancement Tool User's Guide*, document number SC19-1224.

The string *value* is the name of the object type specified in the RULE.

At least one RULE must be defined before a thread will be blocked and canceled, or before the Utility Monitor will be invoked. There is no default value. This element may be defined multiple times within the element DSNUTILB_INTERCEPT.

INCLUDE | EXCLUDE

The child elements INCLUDE and EXCLUDE specify that you want to either include or exclude the targets defined within a RULE or RULESET. This element is required both before and after a RULE or RULESET element. INCLUDE Rules allow an action. EXCLUDE Rules prevent an action. These child elements may

also be defined within the element DB2SYSTEM to further refine conditional requirements.

RULESET NAME="name"

The element RULESET defines the name of a set of rules that you want to include or exclude. The parameter NAME is a label for the set of rules you want to create and is required when using the element RULESET. The name of the Rule is arbitrary, and there is no limit on its length. This element is optional, but either RULE or RULESET must be defined at least once in a Policy. There is no default value. This element may be defined multiple times within the element DSNUTILB_INTERCEPT.

USE_RULESET NAME="name"

The element USE_RULESET is specified with the element DB2SYSTEM and may be specified multiple times. It is optional and specifies which pre-defined sets of rules to include for the specified DB2 subsystem. There is no default value. The rules defined within the NAME will be used as criterion when evaluating threads for blocking and canceling, or utilities for monitoring when the DSNUTILB Intercept is invoked.

USE_PRACTICE NAME="name"

The element USE_PRACTICE is specified within the element DB2SYSTEM and may only be specified once. It is optional and specifies which pre-defined Practice Rule to include for evaluation within the specified DB2 subsystem. There is no default value.

3.2 Sample UET Policy to Invoke the Utility Monitor

The following sample Policy in Example 1 contains rules that will be used by the Utility Monitor. The rules begin and end with the element PRACTICE. Once the rules are defined, the element DB2SYSTEM must be specified to use the ACTION MONITOR_UTILITY; otherwise the Utility Monitor will not be invoked. For a complete UET Policy containing the PRACTICE rules defined in Example 1, refer to Appendix A in this document.

Example 1: Sample portion of a Policy for use with the Utility Monitor.

```
.
.
.
<! -- ----- -->
<! -- DEFINE THE RULES FOR THE UTILITY MONITOR           -->
<! -- ----- -->

<PRACTICE NAME="STANDARDS_1">
  <UTILITY NAME="REORG_TABLESPACE">
    <MONITOR>
      <SYNTAX REMOVE="UNLOAD PAUSE"/>
      <SYNTAX VALUE="SCOPE %" SUBSTITUTE="SCOPE PENDING"
JOURNAL="NO"/>
      <SYNTAX VALUE="LOG NO" SUBSTITUTE="LOG YES"/>
      <SYNTAX VALUE="SHRLEVEL %" SUBSTITUTE="SHRLEVEL NONE"/>
      <SYNTAX VALUE="SORTNUM %" SUBSTITUTE="SORTNUM 12"/>
      <SYNTAX ADD="KEEPDICTIONARY"/>
    </MONITOR>
  </UTILITY>
  <UTILITY NAME="LOAD">
    <MONITOR>
      <SYNTAX VALUE="LOG YES" SUBSTITUTE="LOG NO"/>
    </MONITOR>
  </UTILITY>
```

```

</PRACTICE>
.
.
<DB2SYSTEM SSID="D91A" ACTION="MONITOR_UTILITY">
  <USE_PRACTICE NAME="STANDARDS_1"/>
  <INCLUDE>
    <RULE TABLESPACE="ABPDB1%.ABPQA2%">
  </INCLUDE>
</DB2SYSTEM>
.
.
.

```

The following elements described below are specific to invoking the Utility Monitor and the actions they perform.

PRACTICE

The element PRACTICE identifies a set of specifications that allow utility syntax to be changed at runtime.

NAME="STANDARDS_1"

An attribute of the element PRACTICE that defines the name of the Rule that will consist of subsequent actions. The Name in this example is STANDARDS_1.

UTILITY

The element UTILITY is a child element of PRACTICE that, when used in conjunction with the attribute NAME, identifies the DB2 utility that UET is to evaluate when DSNUTILB is invoked.

NAME="REORG_TABLESPACE"

An attribute of the element UTILITY that defines the name of the DB2 utility that will be evaluated. In this example, if a REORG TABLESPACE utility is run and the subsequent syntax rules match the REORG utility, the Utility Monitor will be invoked.

MONITOR

A child element of UTILITY that identifies a set of syntax rules by which to evaluate the specified DB2 utility.

SYNTAX

A child element of MONITOR that contains options to either add syntax, change syntax, or remove syntax.

REMOVE="UNLOAD PAUSE"

A child element of SYNTAX that will remove a text string. In this example, when the utility REORG TABLESPACE is being executed by DSNUTILB, the parameter UNLOAD PAUSE will be removed if it is present in any REORG TABLESPACE utility. If the parameter UNLOAD PAUSE is not present in the utility syntax, no action is taken.

VALUE="SCOPE %" SUBSTITUTE="SCOPE PENDING" JOURNAL="NO"

A child element of SYNTAX that will substitute a text string based on a wild card pattern. In this example, when the utility REORG TABLESPACE is being executed by DSNUTILB, the parameter SCOPE PENDING will be substituted in the utility if the parameter SCOPE is present in the utility syntax, regardless of what value is

specified with SCOPE. If the parameter SCOPE is not present in the utility syntax, no action is taken. In addition, the action will not be journaled.

VALUE="LOG NO" SUBSTITUTE="LOG YES"

A child element of SYNTAX that will substitute a text string. In this example, when the utility REORG TABLESPACE is being executed by DSNUTILB, the parameter LOG YES will be substituted into the utility if LOG NO is present. Otherwise, if the parameter LOG NO is not present, the specified value is not substituted.

VALUE="SHRLEVEL %" SUBSTITUTE="SHRLEVEL NONE"

A child element of SYNTAX that will substitute a text string based on a wild card pattern. In this example, when the utility REORG TABLESPACE is being executed by DSNUTILB, the parameter SHRLEVEL NONE will be substituted in the utility if the parameter SHRLEVEL is present in the utility syntax, regardless of what value is specified with SHRLEVEL. If SHRLEVEL is not present in the utility syntax, no action is taken.

VALUE="SORTNUM %" SUBSTITUTE="SORTNUM 12"

A child element of SYNTAX that will substitute a text string based on a wild card pattern. In this example, when the utility REORG TABLESPACE is being executed by DSNUTILB, the parameter SORTNUM 12 will be substituted in the utility if the parameter SORTNUM is present in the utility syntax, regardless of what value is specified with SORTNUM. If SORTNUM is not present in the utility syntax, no action is taken.

ADD="KEEPDICTIONARY"

A child element of SYNTAX that will add a text string. In this example, when the utility REORG TABLESPACE is being executed by DSNUTILB, the parameter KEEPDICTIONARY will be added if the parameter is not present. If the parameter already exists, no action is taken.

DB2SYSTEM

The element DB2SYSTEM encompasses the actions and rules with which the specified DB2 SSID will be evaluated.

SSID="D91A"

A child element of DB2SYSTEM that specifies the DB2 SSID on which Policy rules are to be enforced. In this example, the SSID on which DB2 utilities are to be monitored is D91A.

ACTION="MONITOR_UTILITY"

An attribute of element DB2SYSTEM, ACTION specifies the activity that is to occur for the defined subsystem when evaluating the Policy rules. In this example, ACTION contains a value called MONITOR_UTILITY, which will invoke the Utility Monitor for subsystem D91A.

USE_PRACTICE

The element USE_PRACTICE specifies the name of the Practice to use when evaluating the Policy rules.

NAME="STANDARDS_1"

An attribute of the element USE_PRACTICE that defines the name of the Practice Rule that will be used when evaluating the Policy rules. In this example, the Practice

Rule named STANDARDS_1 that was previously defined in the Policy will be used with the specified DB2 subsystem.. The Utility Monitor will then evaluate the Practice Rules defined within the Policy to evaluate if the rules should be invoked.

4 How to Maximize the Benefits of the Utility Monitor

The purpose of the Utility Monitor is to help DBAs ensure that DB2 utilities are not being run with syntax that violate their company standards, or ensure users who should not run particular utilities are prevented from doing so. It can be used as an added layer of security each time a DB2 utility is executed from program DSNUTILB.

Running DB2 utilities in conjunction with a best practices methodology ensures the DBA staff is not paged during off hours for utilities that contained parameters in conflict with company policies. Defining Practice Rules within the UET Policy allows the Utility Monitor to scan the DSNUTILB utility syntax for comparison to the Practice Rules.

Defining Practice Rules by utility name is an efficient manner in which to begin using the Utility Monitor. For example, defining a utility name such as the following ensures that the Utility Monitor will only be invoked when the REORG utility is executing at the TABLESPACE level.

```
UTILITY NAME="REORG_TABLESPACE"
```

Practice Rules can be combined with Rules or Rulesets within the element DB2SYSTEM to further qualify when a Practice Rule is met. For more information on combining Practice Rules with additional Rules or Rulesets, please refer to Appendix A: Sample Policy.

4.1 Example Practice Rules For Use with the Utility Monitor

The following examples help illustrate the flexibility of the Utility Monitor and the control DBA staff and IT Managers can employ. Each of the examples is explained for comprehensiveness and illustrative purposes.

Substituting a Text String for Another

This example focuses on when the element VALUE contains an explicit string that should be matched within the utility before the Utility Monitor is invoked.

In the example below, the Utility Monitor first checks to see if the utility statement represents a REORG TABLESPACE command, as specified by the UTILITY element. If the text string SCOPE ALL is found in the syntax, then the string SCOPE PENDING is substituted in the REORG utility syntax before DSNUTILB executes.

No journal record is written to the UET Journal Tables to log the activity for this syntax rule.

Figure 1: Substituting one text string for another.

```
<UTILITY NAME="REORG_TABLESPACE">
  <MONITOR>
    <SYNTAX VALUE="SCOPE ALL" SUBSTITUTE="SCOPE PENDING"
JOURNAL="NO" />
  </MONITOR>
```

```
</UTILITY>
```

Conditionally Failing a Utility

This example illustrates how to cause a utility to fail if a specific text string is present within utility syntax.

In the example below, the Utility Monitor first checks to see if the utility statement represents a REORG TABLESPACE command, as specified by the UTILITY element. If it does, the Utility Monitor evaluates the second criterion. If the text string LOG YES is found in the utility syntax, then the Utility Monitor will fail the utility, and the utility job step's return code will be that of the value specified with the element FAIL. In addition, as a result of the element FAIL, message ABPU5400E is written to the SYSPRINT data set in the job output indicating to the user why the utility job step failed.

By default, a journal record is written to the UET Journal Tables to log the activity for this syntax rule.

Figure 2: Conditionally failing a utility by syntax parameter.

```
<UTILITY NAME="REORG_TABLESPACE" >
  <MONITOR>
    <SYNTAX VALUE="LOG YES" FAIL="101" />
  </MONITOR>
</UTILITY>
```

In addition to the presence of text strings, other criterion, such as object name or user ID, can be combined with this Practice Rule within the element DB2SYSTEM to fail the utility on specific objects or for specific users. For more information about how to specify additional criterion, please refer to Appendix A: Sample Policy.

In the example below, the element DB2SYSTEM contains additional criterion that will cause a utility to fail only when those criterion are met. Otherwise, UET will not fail the utility.

Figure 3: Conditionally failing a utility by user ID.

```
<DB2SYSTEM SSID="D91A" ACTION="MONITOR_UTILITY" >
  <USE_PRACTICE NAME="FAIL_REORG_STANDARDS" />
  <INCLUDE>
    <RULE UTILITY_USERID="TSOD176" />
  </INCLUDE>
</DB2SYSTEM>
```

Because a user ID is included as additional criterion, UET will only cancel the utility named in the Practice Name 'Fail Reorg Standards' if the user ID submitting the utility is TSOD176. In this manner, you can conditionally fail reorg utilities when the specified criterion is met, instead of all reorg utilities.

For more information about how to specify additional criterion, please refer to Appendix A: Sample Policy.

Unconditionally Failing a Utility

Utilities may be prevented from running based on any combination of elements in a Rule. Some of the criterion that can be used to prevent utilities from running are object type, object name, utility command, parameters present within the utility syntax, user ID, etc.

In the example below, the utility REORG TABLESPACE will be prevented from running and a return code of 1010 will be issued by UET. By default, a journal record is written to the UET Journal Tables to log the activity for this syntax rule.

Figure 4: Unconditionally failing a utility.

```
<UTILITY NAME="REORG_TABLESPACE" >
  <MONITOR FAIL="1010" />
</UTILITY>
```

Or

```
<UTILITY NAME="REORG_TABLESPACE" >
  <MONITOR FAIL="1010" >
  </MONITOR>
</UTILITY>
```

Additional criterion must then be specified within the element DB2SYSTEM to cause the Utility Monitor to be invoked and evaluate the Practice Rule. Otherwise, the Utility Monitor will not fail the utility.

Figure 5: Unconditionally failing a utility based on utility command.

```
<DB2SYSTEM SSI D="D91A" ACTION="MONITOR_UTILITY" >
  <USE_PRACTICE NAME="FAIL_REORG_STANDARDS" />
  <INCLUDE>
    <RULE UTILITY_COMMAND="REORG_TABLESPACE" />
  </INCLUDE>
</DB2SYSTEM>
```

Because the utility command REORG TABLESPACE is included as the criterion on which to invoke the Utility Monitor, DB2 Utilities Enhancement Tool will only fail the utility named in the Practice Name 'Fail Reorg Standards' if the utility being run is a REORG TABLESPACE utility. In this manner, you can unconditionally fail all REORG utilities.

For more information about how to specify additional criterion, refer to "Sample policy used by the DSNUTILB intercept" on page 135.

Substituting a Value in a Keyword/Value Pair

This example demonstrates how to substitute a value when a keyword/value pair is found within the specified syntax. In the example below, the value SORTNUM 10 will be substituted within the utility syntax if the parameter SORTNUM is found, regardless of what value is specified with SORTNUM.

For example, if SORTNUM 2 is specified within the utility syntax, the value SORTNUM 10 will be substituted, and the utility will execute with the parameter SORTNUM 10. If the parameter SORTNUM is not present within the utility syntax, then the value SORTNUM 10 will not be substituted within the syntax, and the utility will execute without the parameter SORTNUM.

Figure 6: Substituting a value when a pattern is matched.

```
<UTILITY NAME="REORG_TABLESPACE" >
  <MONITOR>
    <SYNTAX VALUE="SORTNUM %" SUBSTITUTE="SORTNUM 10" />
  </MONITOR>
```

```
</UTILITY>
```

Removing a Text String From the Utility Syntax

This example demonstrates how to remove a text string from a utility syntax if it is present. In the example below, the parameter UNLOAD PAUSE will be removed from the utility syntax before the utility is executed if the string value is present within the utility syntax. If the text string UNLOAD PAUSE is not present within the utility syntax, no action is taken.

Figure 7: Removing a text string from the utility syntax.

```
<UTILITY NAME="REORG_TABLESPACE">
  <MONITOR>
    <SYNTAX REMOVE="UNLOAD PAUSE" />
  </MONITOR>
</UTILITY>
```

Adding a Text String That Is Not Present

This example demonstrates how to add a text string to the utility syntax if it is not already present. In the example below, the value KEEPDICTIONARY will be added to the utility REORG TABLESPACE if it is not already defined within the utility syntax. Otherwise, if the value KEEPDICTIONARY is already defined within the utility syntax, this Rule will be ignored, and the text string will not be duplicated. The text string is appended to the end of the utility statement only once per SYSIN. If multiple utility commands are listed within one SYSIN, the text string the attribute ADD is only appended to the last utility command of the SYSIN.

Figure 8: Adding a text string that is not present.

```
<UTILITY NAME="REORG_TABLESPACE">
  <MONITOR>
    <SYNTAX ADD="KEEPDICTIONARY" />
  </MONITOR>
</UTILITY>
```

Ensuring a Parameter Is Changed If It Is Present, or Added If It Is Not There

This example demonstrates how to change a value if it is already specified in the syntax, or to add it if it is not there. In the example below, the string WRITE NO will be substituted into the utility syntax if WRITE YES is present. However, if WRITE YES is not present within the utility syntax, then WRITE NO will be added, ensuring that the QUIESCE utility is run with the required values.

Figure 9: Ensuring a text string is either changed or added.

```
<UTILITY NAME="QUIESCE">
  <MONITOR>
    <SYNTAX VALUE="WRITE YES" SUBSTITUTE="WRITE NO" />
    <SYNTAX ADD="WRITE NO" />
  </MONITOR>
</UTILITY>
```

4.2 Journaling Utility Monitor Actions

The actions taken by the Utility Monitor can be written to UET tables, called journaling, in a couple of ways to track the action defined within the Policy and the ultimate changes made to the utility syntax. The attribute JOURNAL can be defined on either the element MONITOR or the child element SYNTAX. Because the default value on both elements is YES, omitting this attribute will cause journaling to occur.

When the attribute JOURNAL="YES" is specified on the element MONITOR or is omitted entirely, all subsequent actions defined in the child element SYNTAX will be journaled, unless JOURNAL="NO" is specified on the child element SYNTAX.

The following two examples illustrate the same journaling behavior. Because JOURNAL="YES" is the default value when it is omitted, the first example causes journaling to take place for the subsequent SYNTAX element.

Figure 10: Default journaling at the utility level.

```
<UT I LI TY NAME="QUI ESCE" >
  <MONI TOR>
    <SYNTAX VALUE="WRI TE YES" SUBSTI TUTE="WRI TE NO" />
  </MONI TOR>
</UT I LI TY>
```

Figure 11: Journaling at the utility level.

```
<UT I LI TY NAME="QUI ESCE" >
  <MONI TOR JOURNAL="YES" >
    <SYNTAX VALUE="WRI TE YES" SUBSTI TUTE="WRI TE NO" />
  </MONI TOR>
</UT I LI TY>
```

The default journaling behavior may be changed by specifying JOURNAL="NO" on either the element MONITOR or the child element SYNTAX. When JOURNAL is specified on the element MONITOR, it takes precedence over the child element SYNTAX, unless JOURNAL is specified on the element SYNTAX, overriding the specification on the parent element MONITOR.

The following syntax will not result in journaling SYNTAX activity. Because JOURNAL="NO" was specified on the parent element MONITOR, this value is used for subsequent SYNTAX child elements.

Figure 12: Journaling at the utility level.

```
<UT I LI TY NAME="QUI ESCE" >
  <MONI TOR JOURNAL="NO" >
    <SYNTAX VALUE="WRI TE YES" SUBSTI TUTE="WRI TE NO" />
    <SYNTAX VALUE="EXCEPTI ONS 5"
      SUBSTI TUTE="EXCEPTI ONS 0" />
  </MONI TOR>
</UT I LI TY>
```

The following syntax will result in journaling only those SYNTAX elements where JOURNAL="YES" is defined. The value JOURNAL="NO" was specified on the parent element MONITOR. Therefore, this value takes precedence for any subsequent SYNTAX elements, unless JOURNAL is specified on the SYNTAX element, overriding the value on the parent element MONITOR.

Figure 13: Journaling at the utility level with an override at the syntax level.

```
<UTILITY NAME="QUI ESCE" >
  <MONITOR JOURNAL="NO" >
    <SYNTAX VALUE="WRITE YES"
      SUBSTITUTE="WRITE NO" JOURNAL="YES" />
    <SYNTAX VALUE="EXCEPTIONS 5"
      SUBSTITUTE="EXCEPTIONS 0" />
  </MONITOR>
</UTILITY>
```

In this example, any time WRITE YES is found and changed to WRITE NO, then the action is journaled. However, when EXCEPTIONS 5 is changed to EXCEPTIONS 0, the action will not be journaled because JOURNAL="NO" is specified on the element MONITOR.

4.3 How Whitespace is Compressed From Utility Syntax

The attributes ADD, REMOVE, VALUE and SUBSTITUTE of the element SYNTAX allow administrators to dynamically change the utility syntax specified in the SYSIN data set. Before the Utility Monitor can effectively search for user-defined text string in the Practice Rule (e.g. VALUE="LOG NO"), the utility syntax is placed into a standard internal format by removing all extraneous white space from the SYSIN. That is to say, whitespace is compressed out of the utility syntax in the SYSIN data set before the Utility Monitor searches for the text string against which to compare the Policy Practice Rule.

The following LOAD utility statement is an example of how whitespace in the Utility Monitor is compressed. The following example utility syntax might appear in the SYSIN data set:

Figure 14: Sample SYSIN in DSNUTILB JCL.

```
//SYSIN DD *
LOAD DATA REPLACE LOG
NO INTO TABLE ...
```

The UET syntax parser will "normalize" the utility syntax by removing whitespace, such that the syntax from the example above would appear as the following, with only one space between parameters:

Figure 15: What the SYSIN looks like after it is normalized.

```
LOAD DATA REPLACE LOG NO INTO TABLE ...
```

In this case, specifying the Practice Rule below with one space would match the utility syntax to invoke the Utility Monitor:

Figure 16: Policy rule that will match normalized SYSIN.

```
<SYNTAX VALUE="LOG NO" ... />
```

However, the Practice Rule below with three spaces would *not* match the utility syntax and would not invoke the Utility Monitor:

Figure 17: Policy rule that will not match normalized SYSIN.

```
<SYNTAX VALUE=" LOG    NO" . . . />
```

Normalizing the utility syntax before it is evaluated against Policy rules makes it easier for users to define the Policy rules to invoke the Utility Monitor.

5 Appendix A: Sample Policy

A sample Policy is displayed below in Example 2. Each set of Rules and Practice Rules is described in greater detail to illustrate how UET evaluates each Rule or Ruleset, and how they affect utility jobs.

For more information on the Policy rules, please refer to the IBM manual entitled *IBM DB2 Utilities Enhancement Tool User's Guide*, document ID SC19-1224.

Example 2: Sample Policy used by the DSNUTILB Intercept.

```
<?XML VERSION="1.0" ENCODING="UTF-8"?>
<!DOCTYPE OPTIONS SYSTEM "DD:DTD(ABPDTDPL)">
<!-- ***** -->
<!-- * -->
<!-- * ABPPLCY * -->
<!-- * IBM DB2 UTILITIES ENHANCEMENT TOOL FOR Z/OS V2.1 (H2AM210) * -->
<!-- * DSNUTILB INTERCEPTION POLICY * -->
<!-- * -->
<!-- * ***** -->
<!-- * -->
<!-- * 5655-T58 * -->
<!-- * (C) COPYRIGHT ROCKET SOFTWARE, INC. 2002, 2010 ALL RIGHTS * -->
<!-- * RESERVED. * -->
<!-- * -->
<!-- * ***** -->
<DSNUTILB_INTERCEPT>
<!-- ----- -->
<!-- DEFINE THE RULES FOR DB2 OBJECTS -->
<!-- ----- -->

<RULESET NAME="ACCOUNTS_PAYABLE">
  <INCLUDE>
    <RULE TABLESPACE="ABPQDB01.DBAP%" />
    <RULE TABLESPACE="HREMPDB1.DBEMP%" />
  </INCLUDE>
</RULESET>

<RULESET NAME="WEEKLY_MAINTENANCE">
  <INCLUDE>
    <RULE TABLESPACE="ABPQDB01.%"/>
    <RULE TABLESPACE="ABPQDB02.%"/>
  </INCLUDE>
</RULESET>

<RULESET NAME="APP_DEV">
  <INCLUDE>
    <RULE TABLESPACE="D91ADBB.ADTS3%" PART="1,4:10,75"/>
    <RULE TABLESPACE="D91ADBC.ADTS%" />
  </INCLUDE>
</RULESET>

<!-- ----- -->
<!-- DEFINE THE RULES FOR USER IDs -->
<!-- ----- -->

<RULESET NAME="CANCEL_THESE_USERS">
  <INCLUDE>
    <RULE UTILITY_USERID="PDGROU%" />
    <RULE UTILITY_USERID="CSJENN%" />
    <RULE UTILITY_USERID="PDRI CK%" />
  </INCLUDE>
</RULESET>

<!-- ----- -->
```

```

<!-- DEFINE THE RULES FOR SHRLEVEL                                -->
<!-- ----->
<RULESET NAME="SHRLEVEL_RULE">
  <INCLUDE>
    <RULE SHRLEVEL="REFERENCE" />
    <RULE SHRLEVEL="NONE" />
  </INCLUDE>
</RULESET>

<!-- ----->
<!-- DEFINE THE RULES FOR DB2 UTILITIES                          -->
<!-- ----->

<RULSET NAME="EXPENSE_REPORTS">
  <INCLUDE>
    <RULE UTILITY_ID="UTID573%" />
    <RULE UTILITY_COMMAND="REORG_TABLESPACE" />
    <RULE TABLESPACE="TRVLDB.EXPRPTS" />
  </INCLUDE>
</RULSET>

<RULESET NAME="PAYROLL">
  <INCLUDE>
    <RULE UTILITY_COMMAND="REORG_TABLESPACE" />
    <RULE TABLESPACE="DBPRO%.%" />
  </INCLUDE>
</RULESET>

<!-- ----->
<!-- DEFINE THE RULES FOR THE UTILITY MONITOR                    -->
<!-- ----->

<PRACTICE NAME="NORMAL_STANDARDS">
  <UTILITY NAME="REORG_TABLESPACE">
    <MONITOR>
      <SYNTAX REMOVE="UNLOAD PAUSE" />
      <SYNTAX VALUE="LOG NO" SUBSTITUTE="LOG YES" />
      <SYNTAX VALUE="SHRLEVEL %" SUBSTITUTE="SHRLEVEL NONE" />
      <SYNTAX ADD="KEEPDICTIONARY" />
    </MONITOR>
  </UTILITY>

  <UTILITY NAME="LOAD">
    <MONITOR>
      <SYNTAX VALUE="LOG YES" SUBSTITUTE="LOG NO" />
    </MONITOR>
  </UTILITY>
</PRACTICE>

  <PRACTICE NAME="FAIL_REORG_STANDARDS">
    <UTILITY NAME="REORG_TABLESPACE">
      <MONITOR FAIL="100" />
    </UTILITY>
  </PRACTICE>

<!-- ----->
<!-- SPECIFY WHICH RULES TO USE ON SPECIFIC DB2 SSI DS          -->
<!-- ----->

<POLICY>

<!-- DB2 SSID - D91A: APPLICATION DEVELOPMENT                    -->

<DB2SYSTEM SSID="D91A">
  <USE_RULESET NAME="SHRLEVEL_RULE" />
  <USE_RULESET NAME="APP_DEV" />
</DB2SYSTEM>

<DB2SYSTEM SSID="D91A" ACTION="MONITOR_UTILITY">
  <USE_PRACTICE NAME="NORMAL_STANDARDS" />
  <INCLUDE>

```

```

        <RULE UTILITY_USERID="%" />
      </INCLUDE>
</DB2SYSTEM>

<! -- DB2 SSID - Q91A: APPLICATION TEST -->

<DB2SYSTEM SSID="Q91A" ACTION="MONITOR_UTILITY">
  <USE_PRACTICE NAME="FAIL_REORG_STANDARDS" />
  <INCLUDE>
    <RULESET NAME="CANCEL_THESE_USERS" />
  </INCLUDE>
</DB2SYSTEM>

</POLICY>
</DSNUTILB_INTERCEPT>

```

In general, rules should be defined from the least specific to the most specific. The rules in the <RULESET> section are read and processed from top to bottom. Therefore, the order in which rules are defined is important.

RULESET NAME="ACCOUNTS_PAYABLE"

Within this Ruleset named ACCOUNTS_PAYABLE, there are two tablespaces being included for evaluation. Because the tablespace names are wild carded, any match on the pattern within a DB2 utility invokes the ACTION specified in the element DB2SYSTEM. In the example below, if a utility is running at the tablespace level (such as REORG TABLESPACE) and it contains a tablespace name whose pattern matches ABPQDB01.DBAP% or HREMPDB1.DBEMP%, then the ACTION specified in the element DB2SYSTEM is invoked.

When two or more Rules of the same type are included in one Ruleset they are OR'd together. That is, only one of these Rules must match in order for the ACTION to be invoked.

```

<RULESET NAME="ACCOUNTS_PAYABLE">
  <INCLUDE>
    <RULE TABLESPACE="ABPQDB01.DBAP%" />
    <RULE TABLESPACE="HREMPDB1.DBEMP%" />
  </INCLUDE>
</RULESET>

```

RULESET NAME="WEEKLY_MAINTENANCE"

Within this Ruleset named WEEKLY_MAINTENANCE, there are two tablespaces being included for evaluation. Because the tablespace names are wild carded, any match on the pattern found within a DB2 utility invokes the ACTION specified in the element DB2SYSTEM. In the example below, if a utility is running at the tablespace level (such as REORG TABLESPACE) and it contains a tablespace name whose pattern matches ABPQDB01.% or ABPQDB02.%, then the ACTION specified in the element DB2SYSTEM is invoked.

When two or more Rules of the same type are included in one Ruleset they are OR'd together. That is, only one of these Rules must match in order for the ACTION to be invoked.

```

<RULESET NAME="WEEKLY_MAINTENANCE">
  <INCLUDE>
    <RULE TABLESPACE="ABPQDB01.%" />
    <RULE TABLESPACE="ABPQDB02.%" />
  </INCLUDE>

```

```
</I NCLUDE>  
</RULESET>
```

RULESET NAME="APP_DEV"

Within this Ruleset named APP_DEV, there are two tablespaces being included for evaluation. Because the tablespace names are wild carded, any match on the pattern within a DB2 utility invokes the ACTION specified in the element DB2SYSTEM. In the example below, if a utility is running at the tablespace level (such as REORG TABLESPACE) and it contains a tablespace name whose pattern matches D91ADBB.ADTS3% and is executing against parts 1, parts 4 through 10, or part 75, or the pattern matches D91ADBC.ADTS%, then the ACTION specified in the element DB2SYSTEM is invoked.

When two or more Rules of the same type are included in one Ruleset they are OR'd together. That is, only one of these Rules must match in order for the ACTION to be invoked.

```
<RULESET NAME="APP_DEV" >  
  <I NCLUDE>  
    <RULE TABLESPACE="D91ADBB. ADTS3%" PART="1, 4: 10, 75" />  
    <RULE TABLESPACE="D91ADBC. ADTS%" />  
  </I NCLUDE>  
</RULESET>
```

RULESET NAME="CANCEL_THESE_USERS"

Within this Ruleset named CANCEL_THESE_USERS, there are three TSO user IDs being included for evaluation. Because the user IDs are wild carded, any match on the pattern found within a DB2 utility will invoke the ACTION specified in the element DB2SYSTEM. In the example below, if a utility is submitted by a TSO user ID whose pattern matches PDGROU%, CSJENN% or PDRICK%, then the ACTION specified in the element DB2SYSTEM is invoked.

When two or more Rules of the same type are included in one Ruleset they are OR'd together. That is, only one of these Rules must match in order for the ACTION to be invoked.

```
<RULESET NAME="CANCEL_THESE_USERS" >  
  <I NCLUDE>  
    <RULE UTI LI TY_ USERI D="PDGROU%" />  
    <RULE UTI LI TY_ USERI D="CSJENN%" />  
    <RULE UTI LI TY_ USERI D="PDRI CK%" />  
  </I NCLUDE>  
</RULESET>
```

RULESET NAME="SHRLEVEL_RULE"

Within this Ruleset named SHRLEVEL_RULES, there are two SHRLEVEL values being included for evaluation. Because the SHRLEVEL values are explicitly defined and not wild carded, the syntax in the utility must match to invoke the ACTION specified in the element DB2SYSTEM.

In the example below, if a utility is submitted with either SHRLEVEL REFERENCE or SHRLEVEL NONE, then the ACTION specified in the element DB2SYSTEM is invoked.

When two or more Rules of the same type are included in one Ruleset they are OR'd together. That is, only one of these Rules must match in order for the ACTION to be invoked.

```
<RULESET NAME="SHRLEVEL_RULE">
<INCLUDE>
  <RULE SHRLEVEL="REFERENCE" />
  <RULE SHRLEVEL="NONE" />
</INCLUDE>
</RULESET>
```

RULESET NAME="EXPENSE_REPORTS"

Within this Ruleset named EXPENSE_REPORTS, there are three Rules containing different values against which the utility will be evaluated.

In the example below, the first Rule specifies a pattern for a DB2 utility ID. The second Rule specifies a DB2 utility command. And the third Rule specifies an object name. If a utility is submitted with a user ID that matches the DB2 utility ID pattern UTID573%, with a REORG TABLESPACE utility, and against object TRVLDB.EXPRPTS, then the ACTION specified in the element DB2SYSTEM is invoked. If any criterion doesn't match within the utility being submitted, then the ACTION defined within the element DB2SYSTEM is not invoked.

When two or more Rules of different types are included in one Ruleset they are AND'd together. That is, all of the Rules within the defined Ruleset must match in order for the ACTION to be invoked.

```
<RULESET NAME="EXPENSE_REPORTS">
<INCLUDE>
  <RULE UTILITY_ID="UTID573%" />
  <RULE UTILITY_COMMAND="REORG_TABLESPACE" />
  <RULE TABLESPACE="TRVLDB. EXPRPTS" />
</INCLUDE>
</RULESET>
```

RULESET NAME="PAYROLL"

Within this Ruleset named PAYROLL, there are two Rules containing different values against which the utility will be evaluated.

In the example below, the first Rule specifies a DB2 utility command, and the second Rule specifies a pattern for an object name. If a utility is submitted with the syntax REORG TABLESPACE and specifies a tablespace name matching the pattern DBPR0%.%, then the ACTION specified in the element DB2SYSTEM is invoked. If any criterion doesn't match within the utility being submitted, then the ACTION defined within the element DB2SYSTEM is not invoked.

When two or more Rules of different types are included in one Ruleset they are AND'd together. That is, all of the Rules within the defined Ruleset must match in order for the ACTION to be invoked.

```
<RULESET NAME="PAYROLL">
<INCLUDE>
  <RULE UTILITY_COMMAND="REORG_TABLESPACE" />
  <RULE TABLESPACE="DBPR0%.%" />
</INCLUDE>
```

```
</I NCLUDE>
</RULESET>
```

PRACTICE NAME="NORMAL_STANDARDS"

Within this Practice Rule named NORMAL_STANDARDS, there are two utilities being monitored, each with their own set of syntax definitions. The NAME defined within the element UTILITY must match the utility submitted in order for UET to monitor the named utility. Each SYNTAX attribute will be evaluated independently, rather than evaluating all SYNTAX attributes as a group. Only one SYNTAX attribute must match in order to perform the action each attribute defines (i.e., ADD, REMOVE, VALUE/SUBSTITUTE).

In the example below, the utility REORG TABLESPACE will be monitored for four syntax strings. The first syntax string specifies to remove the text UNLOAD PAUSE if it is present. The second syntax string specifies to substitute LOG YES instead of LOG NO if LOG NO is present. The third syntax string specifies to substitute the given SHRLEVEL value in the utility syntax with SHRLEVEL NONE. The fourth syntax string KEEPDICTIONARY will be added to the utility syntax.

```
<PRACTICE NAME="NORMAL_STANDARDS">
  <UTILITY NAME="REORG_TABLESPACE">
    <MONITOR>
      <SYNTAX REMOVE="UNLOAD PAUSE"/>
      <SYNTAX VALUE="LOG NO" SUBSTITUTE="LOG YES"/>
      <SYNTAX VALUE="SHRLEVEL %" SUBSTITUTE="SHRLEVEL NONE"/>
      <SYNTAX ADD="KEEPDICTIONARY"/>
    </MONITOR>
  </UTILITY>
</PRACTICE>
```

In the example below, the utility LOAD will be monitored for one syntax string. If the syntax string LOG YES is present in the utility syntax, it will be replaced by LOG NO.

```
<UTILITY NAME="LOAD">
  <MONITOR>
    <SYNTAX VALUE="LOG YES" SUBSTITUTE="LOG NO"/>
  </MONITOR>
</UTILITY>
</PRACTICE>
```

PRACTICE NAME="FAIL_REORG_STANDARDS"

Within this Practice Rule named FAIL_REORG_STANDARDS, there is one utility being monitored. The NAME defined within the element UTILITY must match the utility submitted in order for UET to monitor the named utility.

In the example below, the utility REORG TABLESPACE will not monitor syntax strings. Instead, it will simply fail the utility before it ever starts, preventing it from being run. It will also issue the return code 100, indicating to the user that there was an error within the utility job. When this happens, a message will be issued within the SYSPRINT indicating to the submitter that the job was purposely failed and the reason why.

```
<PRACTICE NAME="FAIL_REORG_STANDARDS">
  <UTILITY NAME="REORG_TABLESPACE">
    <MONITOR FAIL="100"/>
  </UTILITY>
</PRACTICE>
```

DB2SYSTEM SSID="D91A"

The Policy definitions that were previously defined are enforced within the element DB2SYSTEM. The Policy rules are enforced by subsystem. The attribute SSID can be wildcarded, so as to apply to multiple DB2 subsystems.

In this example, there is no ACTION attribute specified, so the default value of BLOCK_AND_CANCEL_TREADS will be used. Two Rulesets are included for enforcement on subsystem D91A: SHRLEVEL_RULE and APP_DEV. Each Ruleset is evaluated independently. That is to say, the Ruleset SHRLEVEL_RULE will be evaluated before the Ruleset APP_DEV is evaluated.

The Ruleset SHRLEVEL_RULE will cause threads to be blocked and canceled on the associated DB2 objects any time SHRLEVEL REFERENCE or SHRLEVEL NONE is specified. In addition, any time a utility is run on the objects associated with the Ruleset APP_DEV, threads will be blocked and canceled.

```
<DB2SYSTEM SSID="D91A" >
  <USE_RULESET NAME="SHRLEVEL_RULE" />
  <USE_RULESET NAME="APP_DEV" />
</DB2SYSTEM>
```

DB2SYSTEM SSID="D91A" ACTION="MONITOR_UTILITY"

The Policy definitions that were previously defined are enforced within the element DB2SYSTEM. The Policy rules are enforced by subsystem. The attribute SSID can be wildcarded, so as to apply to multiple DB2 subsystems.

In this example, the ACTION attribute is specified with the value MONITOR_UTILITY, which will invoke the Utility Monitor. No threads will be canceled within this DB2SYSTEM definition; threads are canceled for this subsystem within the previously discussed DB2SYSTEM element description.

In this example, the Practice Rule called NORMAL_STANDARDS will be invoked only when the Rule specifying a user ID matches the user ID value within a utility job. Because there is a wild card within the Rule UTILITY_USERID, all batch utility jobs will cause the Practice Rule NORMAL_STANDARDS to be invoked.

```
<DB2SYSTEM SSID="D91A" ACTION="MONITOR_UTILITY" >
  <USE_PRACTICE_RULE NAME="NORMAL_STANDARDS" />
  <INCLUDE>
    <RULE UTILITY_USERID="%" />
  </INCLUDE>
</DB2SYSTEM>
```

DB2SYSTEM SSID="Q91A" ACTION="MONITOR_UTILITY"

The Policy definitions that were previously defined are enforced within the element DB2SYSTEM. The Policy rules are enforced by subsystem. The attribute SSID can be wildcarded, so as to apply to multiple DB2 subsystems.

In this example, the ACTION attribute is specified with the value MONITOR_UTILITY, which will invoke the Utility Monitor. No threads will be canceled within this DB2SYSTEM definition; threads are canceled for this subsystem within the previously discussed DB2SYSTEM element description.

In this example Practice Rule called FAIL_REORG_STANDARDS will be invoked only when the criterion contained within the Ruleset CANCEL_THESE_USERS matches the

user IDs associated with utility batch jobs. This Practice Rule allows you to purposely fail a utility if a batch job is submitted by any user ID matching those defined in the Ruleset CANCEL_THESE_USERS. This function provides granularity in ensuring only those persons allowed to run a utility may do so.

```
<DB2SYSTEM SSI D="Q91A" ACTION="MONI TOR_UTI LI TY">  
  <USE_PRACTI CE NAME="FAI L_REORG_STANDARDS" />  
  <I NCLUDE>  
    <RULESET NAME="CANCEL_THESE_USERS" />  
  </I NCLUDE>  
</DB2SYSTEM>
```

5.1 Hints and Tips about Policy Rules

This information will help explain the behavior of the parser, enabling you to gain a better understanding of the Policy, its definitions and their effects

The Order of the Rules

The order that the rules are listed in under the element DB2SYSTEM affects how the Policy blocks and cancels threads. The broadest rules should be defined first, and the more specific rules should be defined last.

For example, a Rule specifying which SHRLEVEL keywords to cancel should be listed within the element DB2SYSTEM before rules that contain explicitly-defined objects on which to block and cancel threads.

RULESETs and RULEs Defined Within Each DB2 Subsystem

It should be noted where multiple RULESETs exist that define the same object type and object value, only one RULESET should be included per DB2 subsystem.

For example, in the sample Policy above, the Rulesets ACCOUNTS_PAYABLE and WEEKLY_MAINTENANCE both specify a Rule against a TABLESPACE containing the value ABPQDB01. Therefore it is recommended that only one Ruleset be defined for inclusion or exclusion in the element DB2SYSTEM. Or, if one Ruleset will be included and the other excluded, it is recommended that the more generic Ruleset WEEKLY_MAINTENANCE be defined first, and the more specific Ruleset ACCOUNTS_PAYABLE be defined last.

Unpredictable results may occur if both Rulesets are defined for either inclusion or exclusion within the same DB2SYSTEM element.

RULESETs and RULEs of the Same Object Type

Rulesets that contain rules of the same object type use an "either / or" logic, whereas Rulesets that contain rules of different object types use "and" logic.

For example, a Ruleset that contains multiple rules of the object type DATABASE uses the "either / or" logic. At least one of the objects within the Ruleset must match the object defined in a DB2 utility in order for the ACTION to be invoked. However, if a Ruleset contains rules of different object types (as is in the subsystem D91A in the sample policy), then all criterion within the Ruleset must match a running utility, since different object types are ANDed together.

Matching Rules or Practice Rules with Delimited Table Names

Quotation marks are handled in a specific way within XML. When using Rules or Practice Rules to match against a delimited table name to invoke an ACTION, the text string **"** must be used. This string indicates to XML that a quotation mark should be used when matching the text string value contained within the element or attribute against the table name in a DB2 utility that is executed at the table level.

For example, if a LOAD utility contains a delimited table name in the syntax, such as "CSJENN" . "ABPTB1" , a Rule or Practice Rule would need to be defined like the following in order to match the delimited object name in the DB2 utility. To define a Rule to match a delimited table name, use the following syntax.

```
<RULE TABLE="&quot; CSJENN&quot; . &quot; ABPTB1&quot; " />
```

The resulting table name within the Rule that will be matched against the table name in a DB2 utility will be "CSJENN" . "ABPTB1" . To define a Practice Rule to match a delimited table name, use the following syntax.

```
<SYNTAX VALUE="&quot; CSJENN&quot; . &quot; ABPTB1&quot; "  
SUBSTITUTE="&quot; CSJENN&quot; . &quot; ABPTB1&quot; " />
```

The resulting table name that will be matched within a DB2 utility will be "CSJENN" . "ABPTB1" . The value that will substituted into the DB2 utility will be "CSJENN" . "ABPTB1" .



© Copyright IBM Corporation 2010

IBM United States of America

Produced in the United States of America

All Rights Reserved

The e-business logo, the eServer logo, IBM, the IBM logo, OS/390, zSeries, SecureWay, S/390, Tivoli, DB2, Lotus and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Lotus, Lotus Discovery Server, Lotus QuickPlace, Lotus Notes, Domino, and Sametime are trademarks of Lotus Development Corporation and/or IBM Corporation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PAPER "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Information in this paper as to the availability of products (including portlets) was believed accurate as of the time of publication. IBM cannot guarantee that identified products (including portlets) will continue to be made available by their suppliers.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

Any references in this document to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
4205 South Miami Boulevard
Research Triangle Park, NC 27709 U.S.A.