



| z/TPFDF V1.1

TPF Users Group Fall 2007 Source Scan Tools

Source Scan Extensibility

Mary Komor
TPF Toolkit Task Force

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

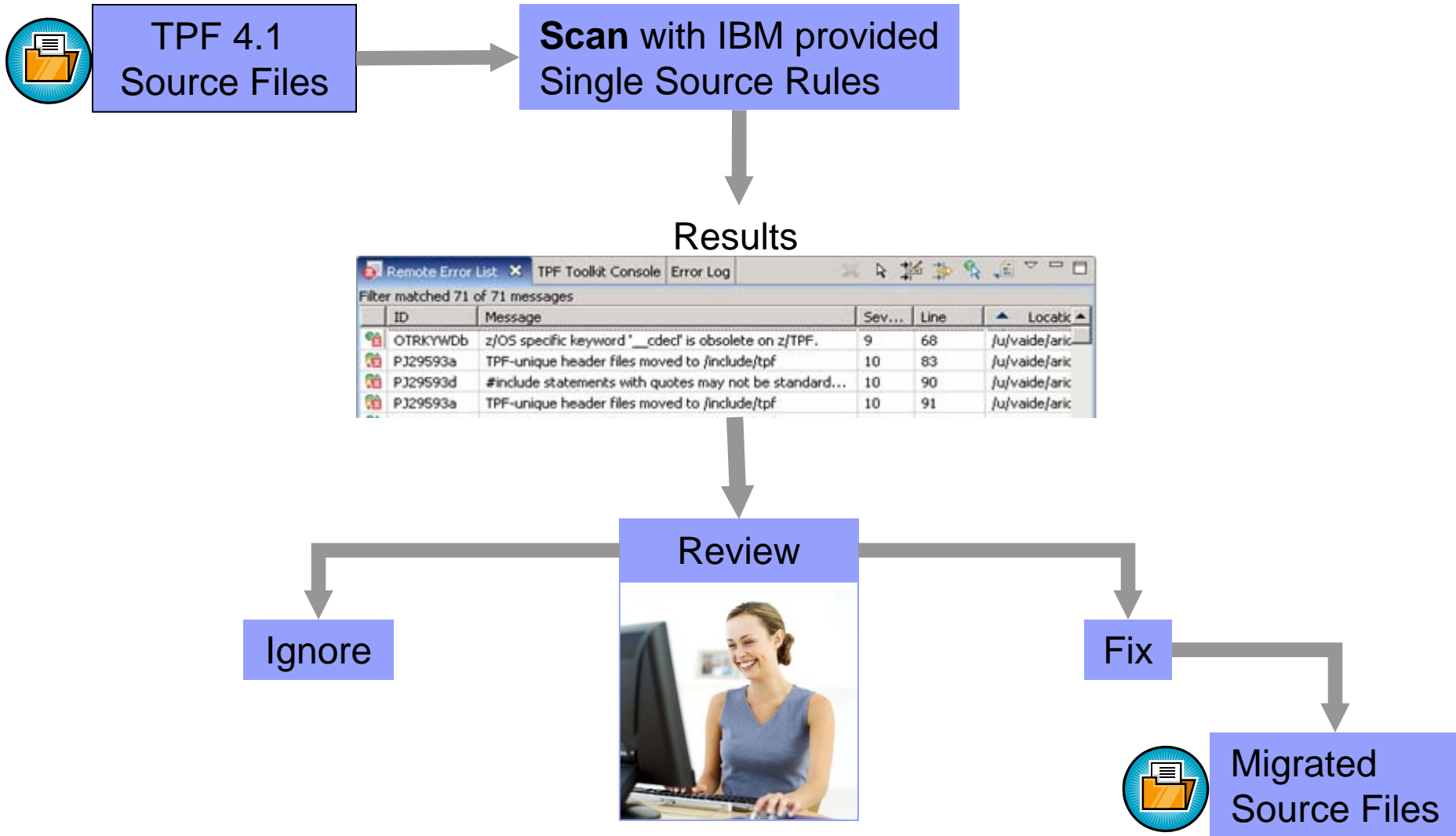
Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

© 2007 IBM Corporation

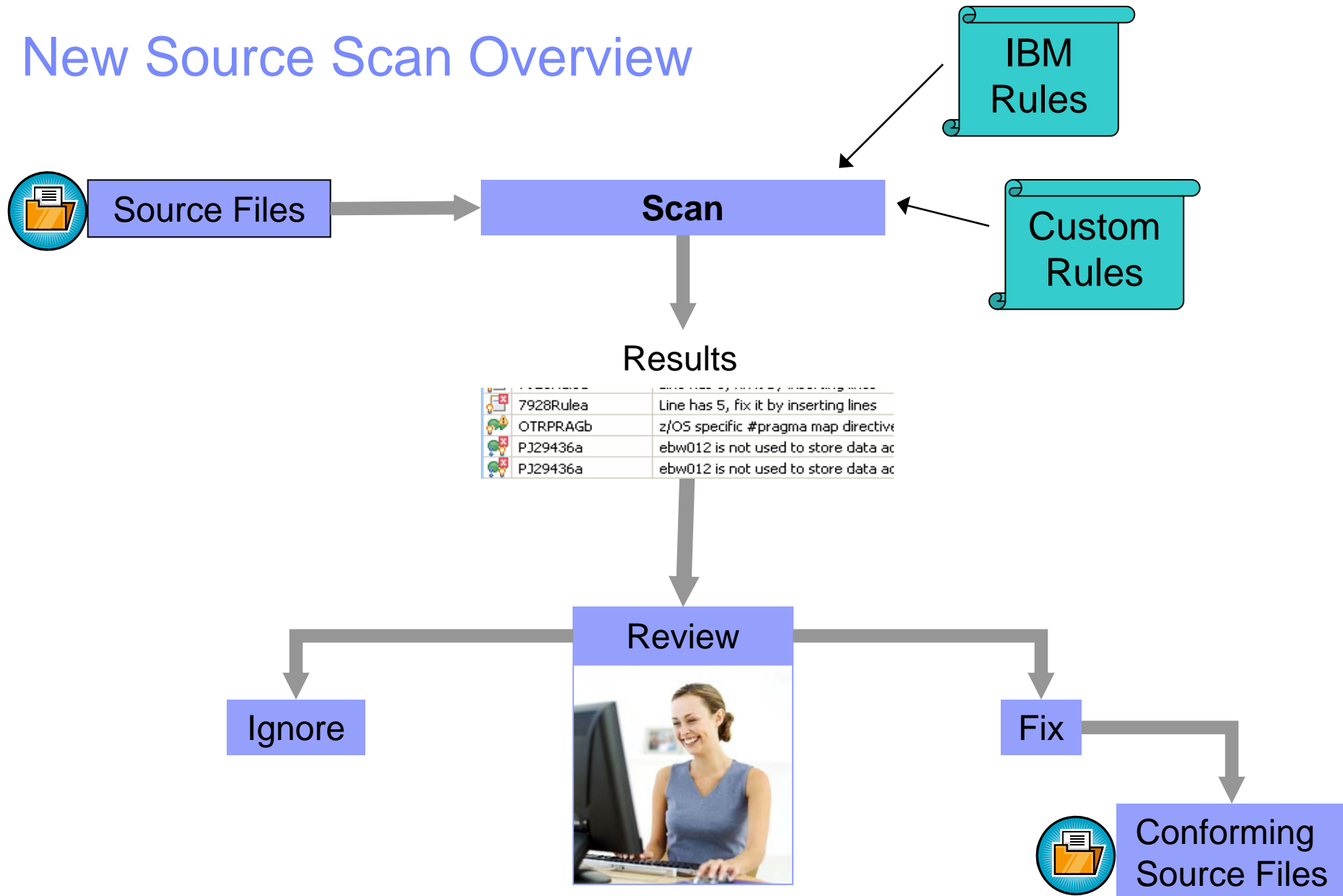
How can you use the source scan tools?

- **Single source migration**
 - Create custom migration rules to migrate application code to be single source compliant
 - Required for z/TPF migration
- **Governance**
 - Enforce enterprise specific coding standards

Current Migration Process Overview

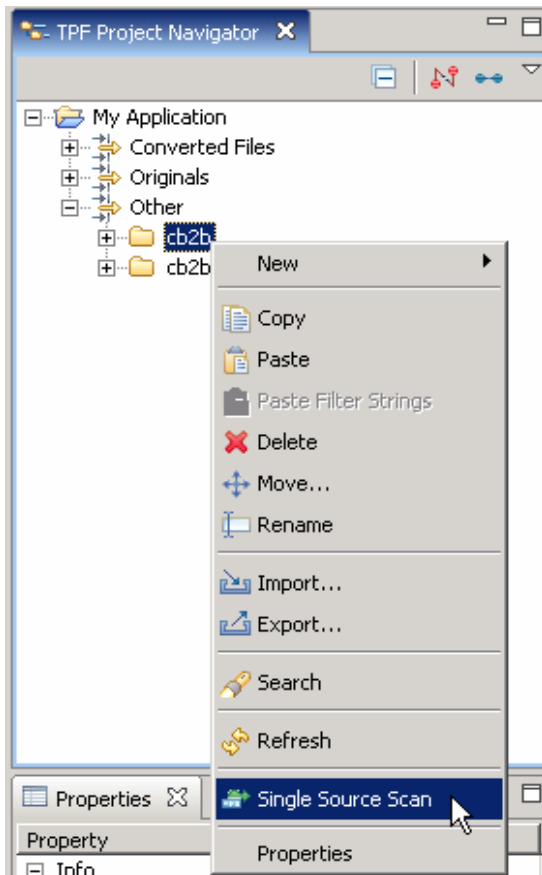


New Source Scan Overview

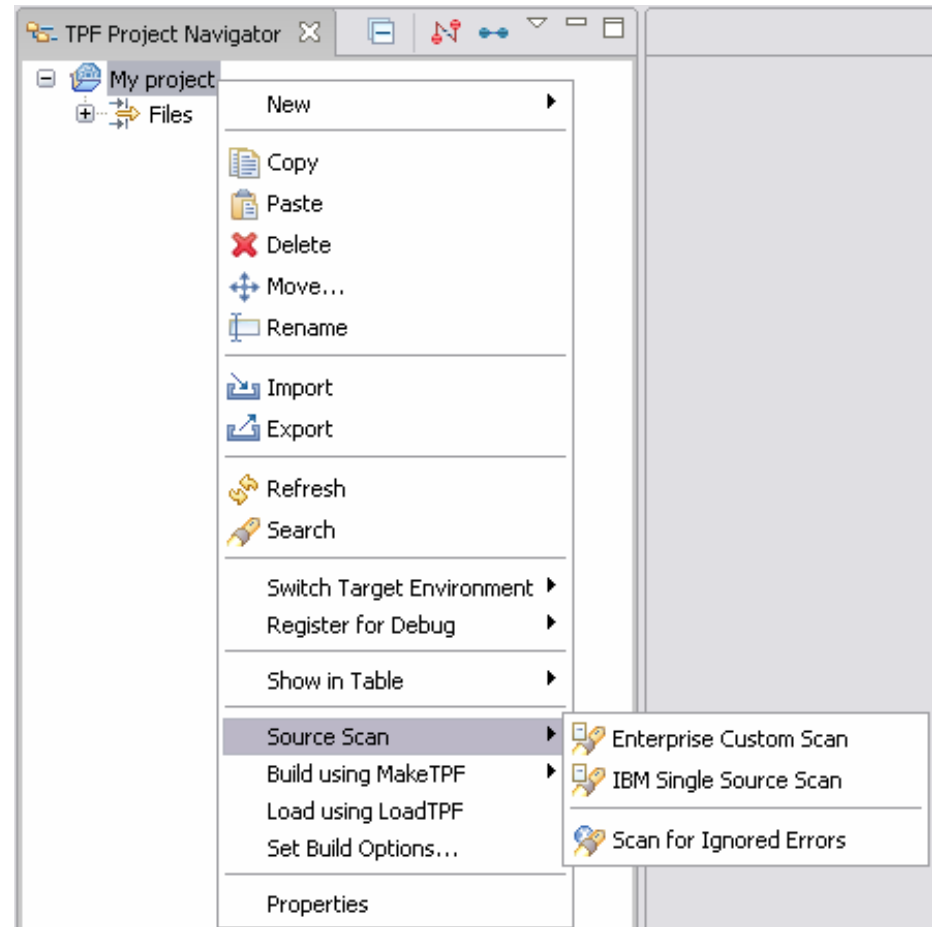


Invoking Scans

BEFORE



NOW



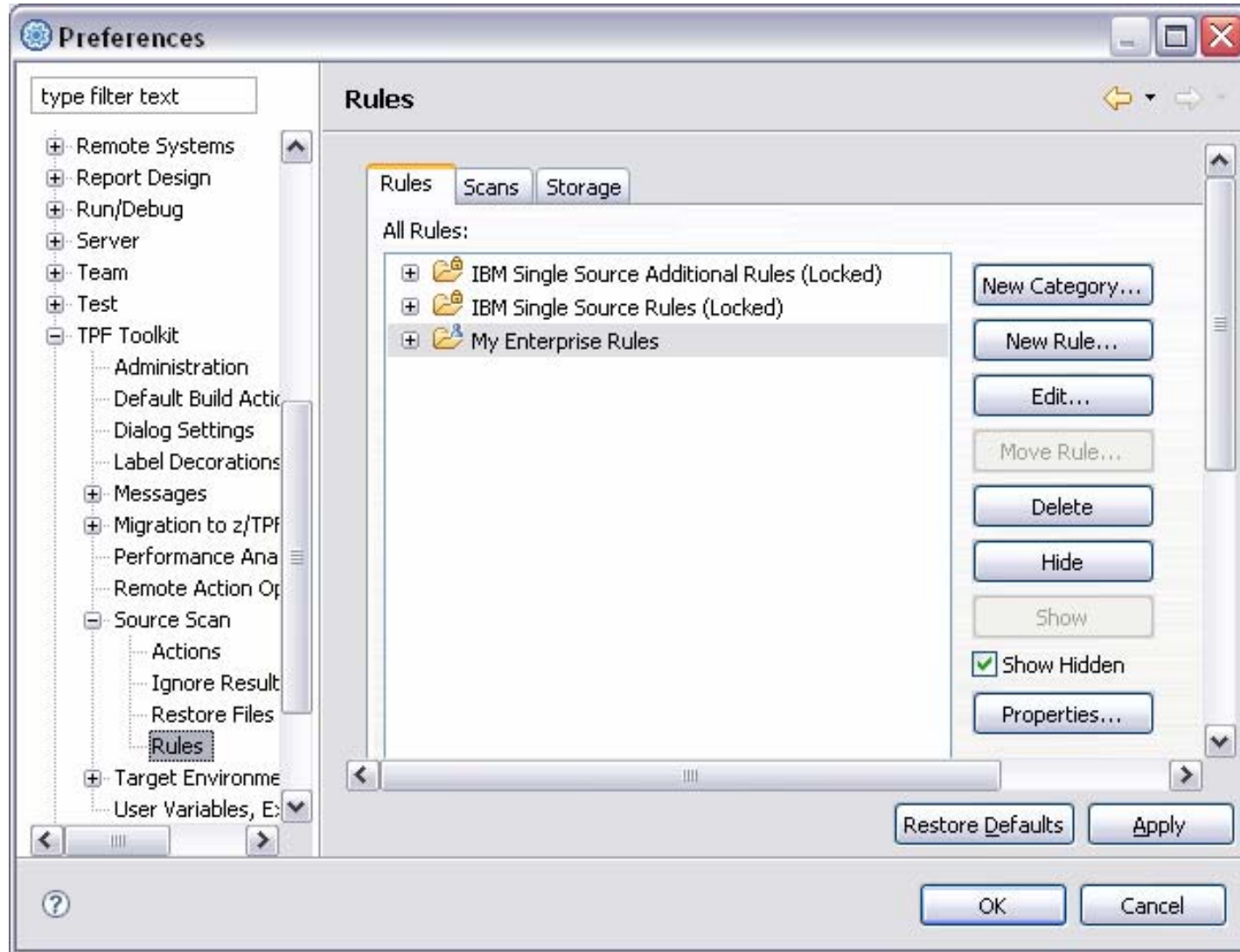
Reviewing Results

- **Types of Results**
 - Potential and Definite
 - Auto Fix and Manual Fix
 - Error and Warnings
 - *NEW – Migration, Performance, General

More About Fixes

- **All fix methods will apply to IBM defined rules as well as user defined rules**
 - Quick Fix in Editors
 - Compare Editor
 - Auto Correct

Source Scan Preference Pages



Templates

- **Detection Templates**
 - Used to describe what the rule will look for
 - C/C++ detection templates
 - ASM detection template
- **Fix Templates**
 - Used to describe how a fix can be made if an error is found
 - C/C++ fix templates
 - ASM fix template

C/CPP Templates

- **CPP Include Statement Rule Template**
- **CPP Plain Text Matching Rule Template**
- **CPP Type Usage Rule Template**
- **CPP Directive Rule Template**
- **CPP Function Call Rule Template**
- **CPP Constant Rule Template**

CPP Include Statement Rule Template

- Only looks within #include directives
- Examples:
 - Flag before.h
 - Flag stdio.h when it is included using angled brackets

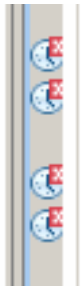
```
#include <iconv.h>           // Character conversion
#include <stddef.h>          // Standard C definitions
#include <stdio.h>           // sprintf
#include <stdarg.h>         // More C stuff
#include <stdlib.h>         // More Character Conversion
#include <string.h>         // String stuff
#include "before.h"
#include <svsotime.h>       // gettimeofday
```

Note: specify file name only or name and quotes/angle brackets

- Single Source example:
PJ29593b - Remove \$ symbol from all header file names

CPP Plain Text Matching Rule

- Looks at all the text in the file excluding comment lines
- Most generic rule - use this template when other more specific templates don't apply
- Example: Flag all usage of the word Wait



```
void SetWaitPrint(BOOL value)
{ printWaitMessages = value; }

void SetWaitInterval(MQLONG WaitInterval)
{ getMessageOptions.WaitInterval = WaitInterval; }
```

Note: This template can flag almost anything

- **Single Source example:**
PJ29436a - OTREXTCa - Upper case "C" required in linkage specification

CPP Fix Templates

- **Replace matched text**
 - Replaces all text matched at the error location with the specified replacement text
 - Examples: Wrap a directive in an #ifdef, Replace one data type name with another data type name.
- **Insert additional line**
 - Inserts a line into the code.
 - Example: Add an include statement
- **Replace matched include name with data file replacement name (*Only available for CPP Include Rule Template)**
 - Similar to the replace matched text fix except that this one will read the replacement value from a data file.

HLASM Detection Template

- **One template matches instructions based on**
 - Label
 - Opcode
 - Positional Operands
 - Keyword Operands

Detection Criteria

Describe the instruction or macros you want to match

Label

Note: Wildcards and regular expressions can only be inserted using 'Insert Pattern'

Opcode

Note: Wildcards and regular expressions can only be inserted using 'Insert Pattern'

Positional Operands

Combine using:

And

Or

And Or

Keyword Parameters

Combine using:

And

Or

HLASM Fix Template

- **One fix template**
 - Modify the matched opcode
 - Modify operands

Choose fix:

Fix Customization

Fix description:

Replace opcode with:

Change Operands

Replace all operands

Replacement text:

Change some operands

Example Rule

- **Requirements**
 - Find C/CPP files that include before.h and replace the file name with after.h

Choosing a Template

- **Best: CPP Include Statement Rule Template**
- **Possible: CPP Directive Rule Template**
 - Provides accurate matching but involves more work to define the match criteria since we have to describe the #include directive as well as the file name and the quote / angle bracket syntaxes.
- **Other: CPP Plain Text Matching Rule Template**
 - Will not scope the search to #include statements



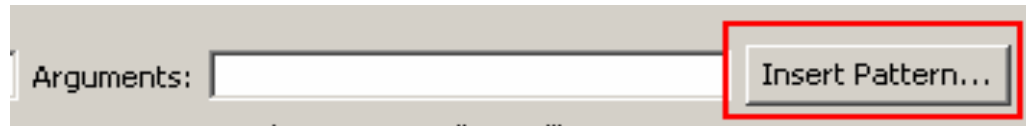
```
if (objectbefore.height == MAX)
{
    defrcount++;
    oute("QMZO3 : No replies received...");
}
```

Advanced Concepts

- **Regular Expressions**
- **More General / More Specific**
- **Preconditions for Compound Rules**
- **Scope Limiting Template**

Regular Expressions

- **Helps you describe more complicated detection text**
- **Regular expression supported in many text fields in templates**
- **Store regular expression as named patterns for re-use**



- **The value matched by the pattern can be used in fixes or in error messages**

More General or More Specific

- Prevent multiple similar rules from flagging the same location
- Method for overriding rules that you did not create
- Example:

Rule A: Flag all include file names that contain a \$ symbol in the name. Manual fix required to rename header file and update include statement as appropriate.

Rule B: Flag include statements that reference my\$file.h and replace it with my_file.h

Problem: The following statement is flagged by both rules

```
#include "my_file.h"
```

Rule B is a better match

- Declare Rule B to be "More Specific" than Rule A
- OR
- Declare Rule A to be "More General" than Rule B

Preconditions

- **Can be used to check other conditions within the file containing the error location.**
- **Rules can use 0 or more preconditions**

Rule: Match calls to function hello()

Precondition: File also includes the helloworld.h header file

Scope Limiting Template

- **A template that is primarily useful for creating preconditions.**
- **When used as a precondition, it will ensure that the rule is matched only when it is within a defined scope.**
- **Examples:**

Rule: Match declarations of char* variables

Precondition: Match only when declaration is within a structure declaration

Example with Precondition (OTRPRGCa)

- **Rule Requirements:**

- **Detection**

This rule flags progc calls that use:

PAT_PBI

PAT_DBI

For example:

```
pat_slot = progc(bklcEnt_ptr->_bklcnam, PAT_PBI);
```

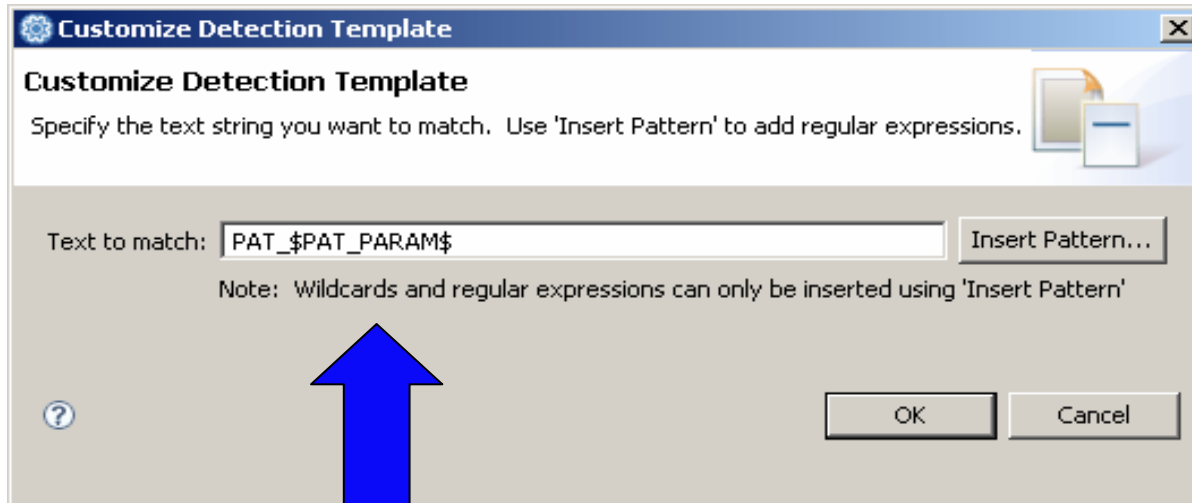
- **Fix**

- PAT_PBI is replaced with PROGC_PBI
 - PAT_DBI is replaced with PROGC_DBI

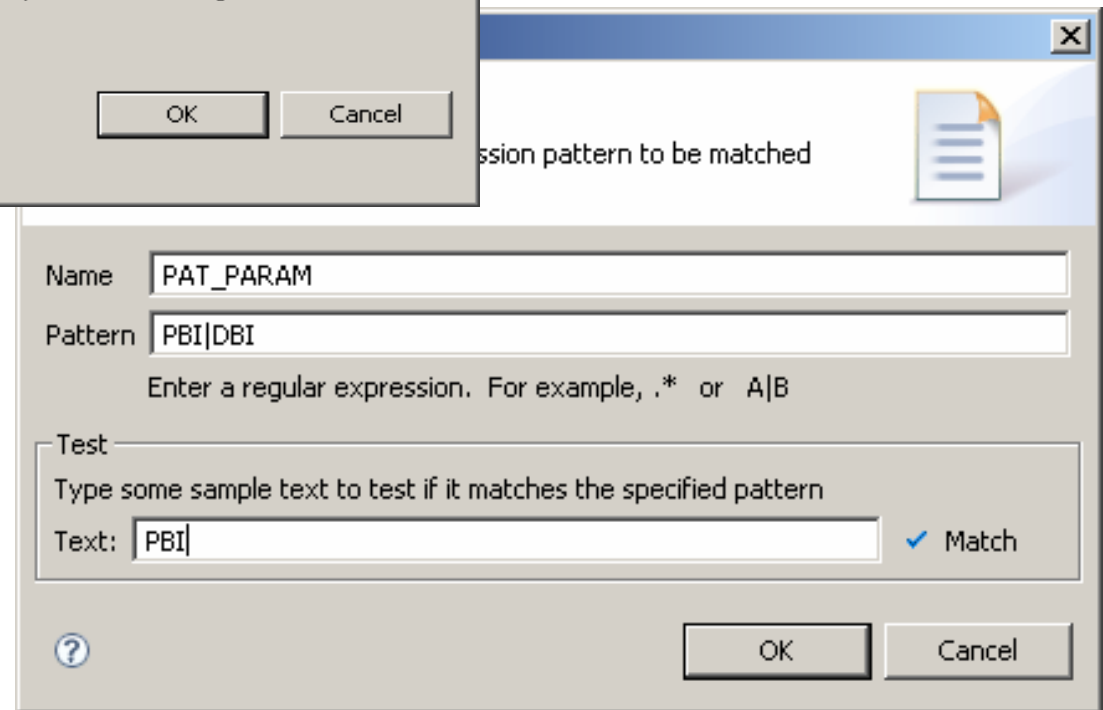
Example with Precondition (OTRPRGCa)

- **Choosing a template**
 - Where is the actual error here?
 - What template will allow us to best describe the error location?
- **Template: Plain text matching**
 - Allows matching text anywhere in a source file. We can use a regular expression to match PAT_PBI or PAT_DBI

Example with Precondition (OTRPRGCa)



Isolate the variable portion of the string so we can re-use this piece in the fix.



Example with Precondition (OTRPRGCa)

- **Because we only care about matching PAT_PBI and PAT_DBI values that are within progc function calls, a precondition can be used to limit the scope**
- **The “CPP Scope Limiting Template” can be used to restrict the scope**

Example with Precondition (OTRPRGCa)

Edit Precondition

Update the precondition information and match criteria

Name:

Location:

Precondition is satisfied when:

Matched (e.g. File contains #include "file.h")

Not matched (e.g. File does not contain #include "file.h")

Detection Criteria

Existing rule

Existing rule:

New Condition:

Template:

Storage File

File:

Precondition Detection Criteria

Specify a scope

Contained within:

Structure

Call to function:

Note: Wildcards and regular expressions can only be inserted using 'Insert Pattern'
Note: Leave blank to match within calls to any function

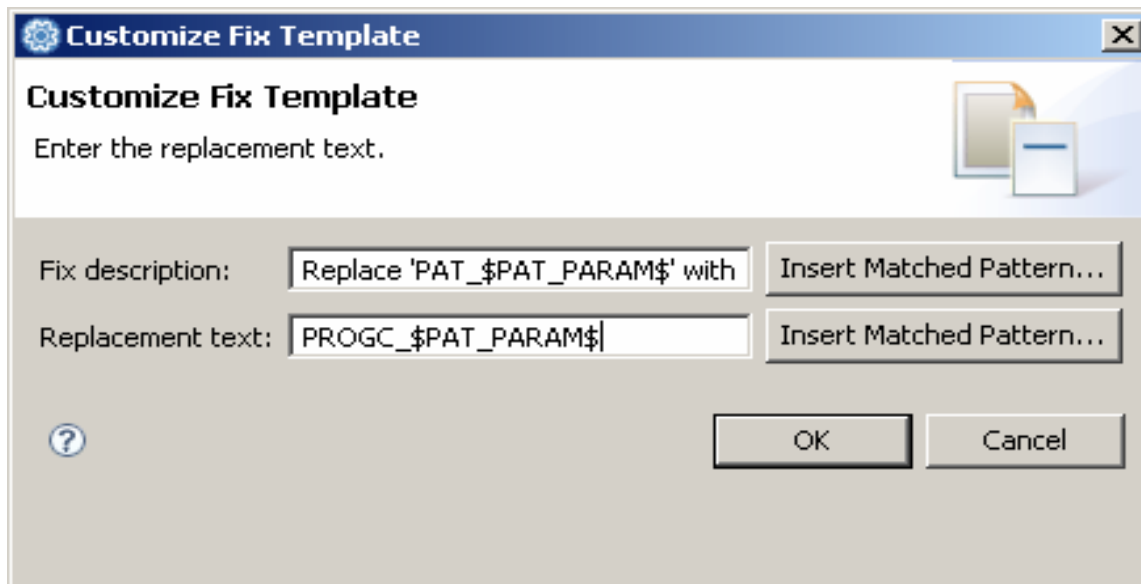
Directive: Arguments:

e.g. ifdef e.g. __370__

Note: Wildcards and regular expressions can only be inserted using 'Insert Pattern'
Note: Leave blank to match any directive or any arguments

Example with Precondition (OTRPRGCa)

- **The fix is to replace the matched text**



Example with Precondition (OTRPRGCa)

```
// OTRPRGCa (Definite, Fixable)
/* pat_slot = progc(bkicEnt_ptr->_bkicnam, PAT_PBI); */
pat_slot = progc(bkicEnt_ptr->_bkicnam, PROGC_PBI);
/* pat_slot = progc(bkicEnt_ptr->_bkicnam, PAT_DBI); */
pat_slot = progc(bkicEnt_ptr->_bkicnam, PROGC_DBI);

// OTRPRGCa (NOT FLAGGED)
pat_slot = notprogc(bkicEnt_ptr->_bkicnam, PAT_PBI);
pat_slot = someotherfuncbkicEnt_ptr->_bkicnam, PAT_DBI);
```

Administration

- **Goals**

- Make it possible for administrators to prevent users from removing and modifying things
- Make it easy to create new rules without having to learn and understand how things are stored
- Make it possible for administrators to use, remove, add and modify rule and scan definitions created by IBM and other administrators without modifying the files created by other people.

Administration Concepts

- **Admin vs User**
 - The tools are aware of whether or not the current user is an administrator or not. If the user is an administrator, they will have permissions that end users don't have
- **Links**
 - Links are used to Add, Hide, and Disable items. The links allow sharing of definitions without modifications of files

Administration Concepts (Cont.)

- **Default Storage File**
 - The file %TPFPROJ%\sourcescandata.xml is reserved as a default storage file
 - Definitions you want to deploy should not be stored in the default storage file
- **File Permissions**
 - When administrators deploy files to end users, they can set permissions on the file that prevents the users from disabling rules in the file

Administration Preference Page

Preferences

type filter text

- [-] General
- [-] Agent Controller
- [-] Analysis
- [-] Ant
- [-] Auto Comment
- [-] Connectivity
- [-] Data
- [-] Help
- [-] Install/Update
- [-] Internet
- [-] Java
- [-] LPEX Editor
- [-] Menu Manager
- [-] Plug-in Development
- [-] Profiling and Logging
- [-] Remote C/C++
- [-] Remote Systems
- [-] Report Design
- [-] Run/Debug
- [-] Server
- [-] Team
- [-] Test
- [-] TPF Toolkit
 - Administration**
 - [-] Default Build Actions
 - [-] Dialog Settings
 - [-] Label Decorations
 - [-] Messages

Administration

Preference Pages | Action Files | Source Scan Files

IBM Source Scan Files

Install	File Name
<input checked="" type="checkbox"/>	%TPFSHARE%\IBMSingleSourceScanData.xml

Edit Permissions...

Enterprise Source Scan Files

File Name	Created By

Remove

Edit Permissions...

Personal Source Scan Files

File Name	Created By
%TPFPROJ%\sourcescandata.xml	arice
%TPFPROJ%\customerdemo.xml	arice

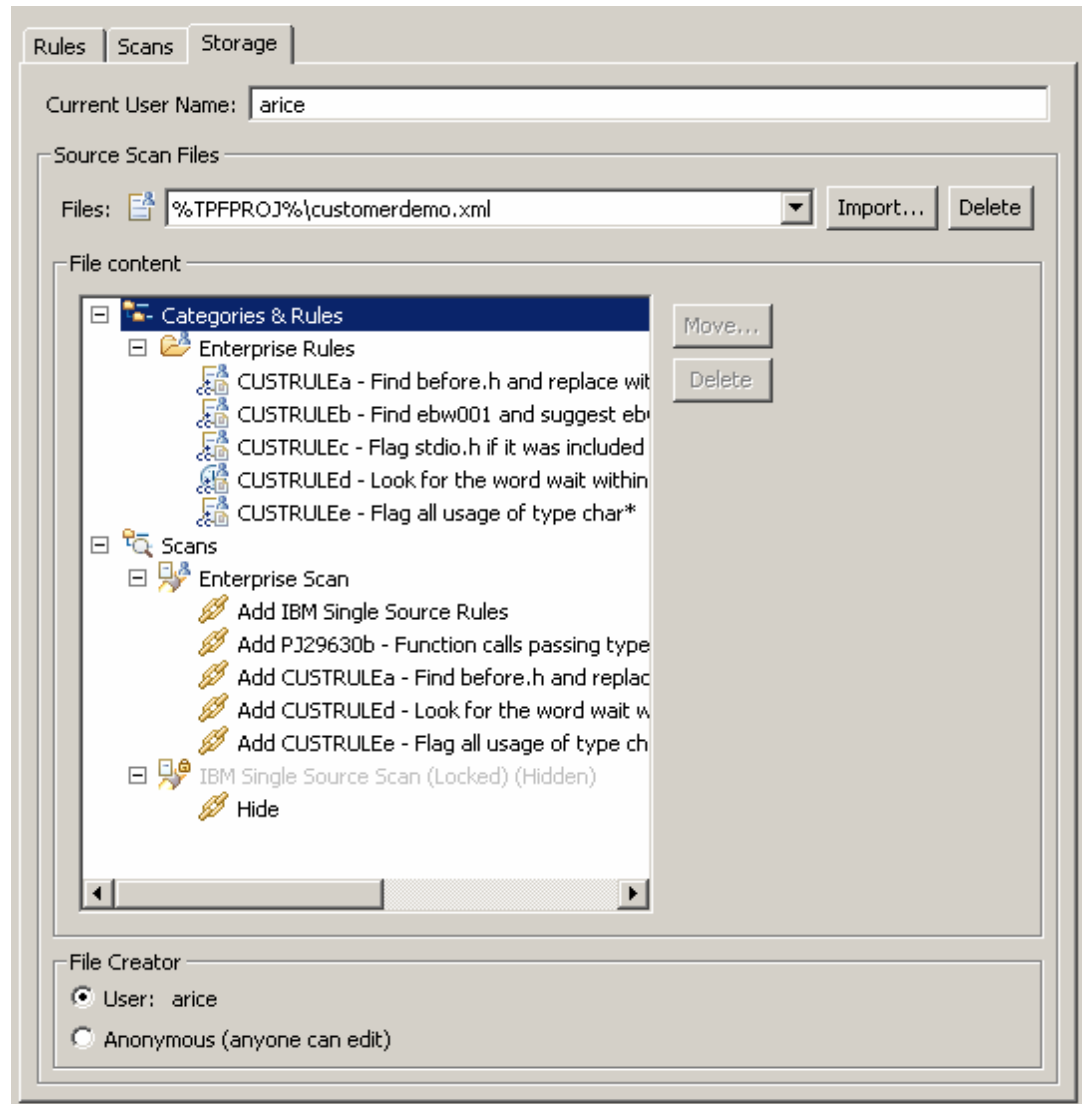
Promote to enterprise

Remove

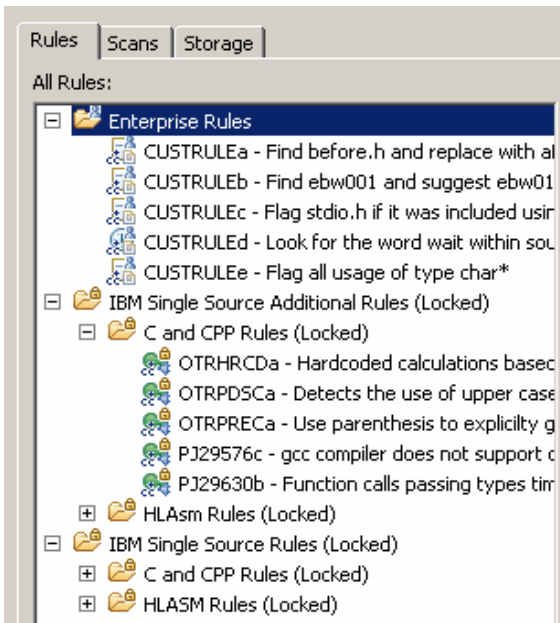
Administrator Editing Options

- Prevent modifications to files created by other users and administrators
- Do not allow any administrator actions while editing (Enforce all permissions)

Storage Tab



Rules



Storage Using Links

File: IBM

Definitions for all IBM provided single source rules, scans and categories.

File: Admin: John

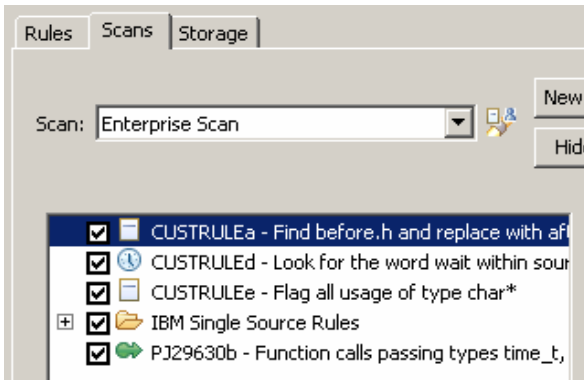
Category Definition: Enterprise Rules

Rule Definition: CISTRULEa [Parent: Enterprise Rules]

Rule Definition: CISTRULEb [Parent: Enterprise Rules]

Rule Definition: CISTRULEc [Parent: Enterprise Rules]

Scans



File: Admin: Sandra

Scan Definition: Enterprise Scan

Rule Definition: CISTRULEd [Parent: Enterprise Rules]

Rule Definition: CISTRULEe [Parent: Enterprise Rules]

Link: Hide Scan: IBM Single Source Scan

Link: Add Category: IBM Single Source Rules to Enterprise Scan

Link: Add CISTRULEa to Enterprise Scan

Link: Add CISTRULEd to Enterprise Scan

Link: Add CISTRULEe to Enterprise Scan

Link: Add PJ29630b to Enterprise Scan

Template vs. Plug-in

- **Currently all IBM rules are provided as plug-in rules. The plug-in rules are hard-coded using Java plug-ins**
- **Customers must create all custom rules using the provided templates**

Questions

- ?
- ?
- ?
- ?
- ?
- ?
- ?

Trademarks

- **IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.**
- **Other company, product, or service names may be trademarks or service marks of others.**
- **Notes**
- **Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.**
- **All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.**
- **This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.**
- **All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.**
- **Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.**
- **Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.**
- **This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.**

Backup slides

CPP Type Usage Rule

- Examines text only when it is being used as a type name
- Example: Flag all `char*` as return type, parameter type, declaration type, cast type

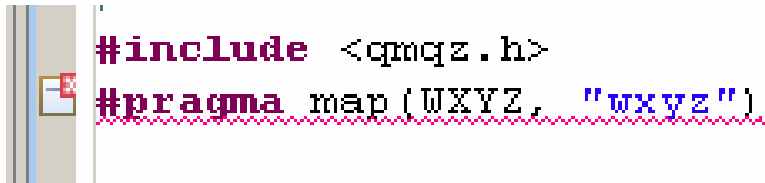
```
char *sf(char *ToOutPut, char *s, int i)
{
    char* buffer = (char*) malloc(strlen(ToOutPut) + 1);

    memcpy(buffer, ToOutPut, strlen(ToOutPut) + 1);
    sprintf(outputbuff, buffer, s, i);
    free(buffer);
    return outputbuff;
}
```

- Single Source example:
OTRWDCTa - Use wide characters (`wchar_t`) for in-memory processing only

CPP Directive Rule Template

- Looks within directives only
- Example: Flag all usage of #pragma

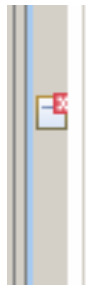


```
#include <qmqz.h>
#pragma map(UXYZ, "xyz")
```

- Single Source Use Case:
OTRPRAGa - Replace or remove #pragma directives

CPP Function Call Rule Template

- Looks for calls to a specified function
- Example: Calls to function named `outc`



```
default:  
    outc(sf("unknown structure type %d", Type));  
    ParameterCount = 0;  
    break;  
}
```

- Single Source example:
PJ29957a - `setlocale` function changed for obsolete category `LC_TOD` (Flags all calls to `setlocale` function)

CPP Constant Rule Template

- Looks only within constants (i.e. string literals and numbers)
- Example: Find MQCLOSE in constants

```
    sprintf(Output, "MQCLOSE(%s, handle %d)",  
              TrgtQName[(H - 1) % queues], H);  
    if (HelpCode(Output)) return;  
  
    MQLONG CompCode, Reason;  
  
    MQCLOSE(Hconn, &Hobj[H - 1], 0, &CompCode, &Reason);  
    CheckError("MQCLOSE", Reason);
```

- Single Source example:
OTRWDCTb - Wide characters coded using hex values break single source