TestBench for IBM i

Feature Summary

Original Software

CONTENTS

© Original Software

DATA EXTRACTION

Data Extraction and sub-setting is provided by the TestBench Data Case component. A Data Case contains information about the relationship between files in the libraries, based on matching data such as key fields, and the selection and sampling criteria defined for the extraction.

Modes

- Extract
- Alter
- Extract and Alter
- Archive

Extract

Enables the extraction of data from a source library, or set of libraries, to a target library, or set of libraries. The data required can be defined by **selection** and **sampling**. Selection extracts data based on selection criteria, similar to an SQL SELECT statement. Sampling extracts data based on combinations of data, with two or more fields being used for sampling.

Alter

Enables the update of data contained in the target library, or libraries. Selection criteria can be used to define which data requires altering. Useful for creating data that currently does not exist.

Extract and Alter

A combination of the above two modes with data being altered as it is extracted. This only affects data in the target library and leaves the source data untouched.

Archive

An Archive is similar in function to an Extract but in this case the selected data is removed from the source library, or libraries, and placed into the target library, or libraries.

Features

- Local Extract – extract from a database that resides on the same partition as the target database
- Remote Extract – extract from a database that resides on a different partition from the target database.
- Auto-Analysis – Automatic definition of database relationships based on field names and access paths.
- Automatic creation of target database – ‘Cloning’ of source database to create the target database.
- Automatic Trigger and Constraint Handling- Management of triggers and constraints in the target database to ensure correct execution of the extract
- Automatic duplicate record avoidance – Prevention of inserts of duplicate records into target database using key field recognition. Provides mechanism for handling un-keyed and non-unique keyed data
- Advanced selection – Selection based on substring or concatenated values. Selection based on calculated values. Selection based on comparison with other data.
- Referential Integrity of selected data – Ensures no ‘orphaned’ data.
- Cascading data selection- Selection and Sampling specified for a parent file is also applied to all child/grandchild files.

© Original Software

- Multiple execution modes – Can be executed interactively, in batch or as a scheduled, submitted job
- Integration with other TestBench components – Can be executed as part of another TestBench component execution e.g. can be run prior to executing a scrambling routine.
- User Exit points – User exit points at every stage of an extraction to enable custom functions to be performed, such as revoking and granting permissions to the target database.
- Independent ASP Support – Data can be extracted from and inserted to databases on IASPs.

DATA SCRAMBLING

Data Scrambling is provided by the TestBench Warp Case component. A Warp Case contains information about the files and data that requires scrambling along with the scrambling method to be applied. Additionally, a Warp Case can be used to manipulate date information in a database to 'move' a database in time, either backwards or forwards.

Features

- Simple Scrambling – scrambling of data in a single file or a list of unrelated files
- Synchronised Scrambling – scrambling of similar data across multiple files to ensure consistent scrambling e.g. applying the same scrambled value to a customer name that appears in three files.
- Data Disassociation method – moving data within a file to effectively de-identify a piece of information such as a Customer Name and Address record.
- Sequential Data method – applying an incrementing value as a scrambled value to a field
- Random Data method – applying a pseud-random generated value as a scrambled value to a field
- Custom Scrambling routines – the ability to apply a custom-defined scrambling routine as a scrambled value to a field. Used when the scrambled data needs to meet certain criteria such as a modulus-10 or -11 check digit. Particularly useful for scrambling National ID numbers and credit card numbers, as an example.
- Advanced Data Scrambling – scrambling of substring and concatenated data.
- Multiple execution modes – Can be executed interactively, in batch or as a scheduled, submitted job
- Integration with other TestBench components – Can be executed as part of another TestBench component execution e.g. can be run after executing a data extraction routine.
- User Exit points – User exit points at every stage of an extraction to enable custom functions to be performed, such as revoking and granting permissions to the target database.
- Independent ASP Support – Data can be scrambled in databases on IASPs.

© Original Software

DATA PROTECTION

Data Protection is provided by the TestBench Test Environment component. A Test Environment contains details of the set of libraries to be considered as a Data Environment and provides a checkpoint & rollback mechanism to protect test data.

Features

- Environment definition – one or more data libraries
- Auto Expanding – new files added to an environment automatically protected
- Checkpoint – a 'point in time' marker to enable TestBench to 'remember' the state of a database
- Rollback – returning a database to the state it was when the specified checkpoint was set
- Extended Rollback Support – support for file actions, such as CLRPFM and RMVM, that are not fully recorded in journals
- Runtime Protection – special 'sandbox' mode to enable repeated testing of an application function with the same data

© Original Software

TESTING CAPABILITIES

IBM i program testing is provided by the TestBench Test Case component. A Test Case is a test harness that contains information about the application or program under test and provides detailed diagnostic information about a test execution. The Test Case enables testing of screen-based programs as well as non-screen based, batch processes. Test Cases can be configured for functional, system regression testing and unit level tests, making the testing capabilities usable by both developers and testers.

Test Case Configuration Features

- Process – the program or command to test. Multiple processes may be defined in a Test Case to facilitate testing of business processes.
- Job date – override of job date to a specified date
- Data Protection Mode – use of TestBench data protection features.
- Job Description – override of job description
- Control Options
    - Capture Spooled Files
    - Intercept Program Calls
    - Log Data Queue Send/Receive
    - Data Queue Protection – restore data queue to pre-test state on completion of the test
    - WebSphere MQ Intercepts – log MQ sends and receives
    - Database Activity Analysis – capture database effects
    - Job log analysis – intercept job log messages
    - Automatic result compare
    - Track submitted jobs

Diagnostic Features

- Joblog – capture of joblog messages related to the test
- Programs – list of programs called (screen-based, non-screen based and batch) showing parameters passed
- Database Effects – list of files updated with details of each record inserted, deleted or updated
- Data Areas – list of data areas updated
- Data Queues – details of messages sent to and received from data queues
- MQ Queues – details of messages sent to and received from MQ queues
- CPU usage – details of CPU milliseconds used by each program
- Data Rules – the application of rules to database effects to ensure data changes are valid
- Spool files – capture of spool files generated during the test

Advanced Testing Features

- Program Interception – monitor for a specific program call, block that call and simulate a return from the called program
- Joblog comparison – compare joblog messages with pre-defined baseline
- Database effects comparison – detailed comparison of file, record and field level database effects with pre-defined baseline
- Spool File compare – rule-based comparison of spool file output with pre-defined baseline

© Original Software

Test Results

Dependent upon the configuration of the Test Case, a test execution result set will contain some or all of the following information:

- Overall Run Status
- Status for each Test Case Process
- CPU mSec for each Test Case Process
- Job log messages
- Program calls with parameters passed in and out of the program
- Database effects – detailed breakdown of every database Write, Update and Delete
- Data Rules – data rules that have been applied to each database effect with details of the pass/fail of the rule
- Data areas – detailed breakdown of changes made to data areas
- Data queues – log of all messages sent to and received from data queues
- MQ Queues – log of all messages sent to and received from MQ Queues
- Spool files – a list of all spool files generated by the test with, optionally, capture of the spool file for review. Captured spool files will also include details of spool file comparison results.
- Timeline – a combination of all the information above, in timestamp sequence, providing a detailed understanding of the test execution

Other Features

- Track Submitted Jobs – monitoring of jobs submitted from the main test. All result components apply to results captured from a submitted job
- Batch Process Monitoring – monitoring of large batch processes. This may include direct program calls and submission of jobs.
- Subsystem job monitoring – monitoring of jobs executed as pre-start job entries

© Original Software

OTHER FEATURES

Impact Analysis

Impact analysis is provided by the TestBench Plan Case component. A Plan Case examines an application (program and files libraries) and builds a cross reference between all the objects. This cross reference provides details of call stacks and called-by stacks and a detailed reference of object relationships. A sophisticated search engine enables users to quickly determine the potential impact of changing an application object whether that object be a program, a service program, a physical or logical file or a message queue.

File Compares

File comparison is provided by the TestBench Compare Case component. Files of the same structure or different structures can be compared as can fields of differing data types.

User Exit Points

Throughout TestBench, user exit points are provided to enable the user to extend the capabilities of the TestBench components. For example, a user exit may be used to revoke authority to a data library prior to refreshing the data using a Data Case. Once the data library has been refreshed, and scrambled using a Warp Case, a final user exit can be used to grant authorities again.

© Original Software