

CENTRE™ (Common Enterprise Resource)

CENTRE Interfaces to external Applications

There are several subsystems within CENTRE where a need had risen to exchange data between CENTRE and other third party software applications. Such subsystems include Time Sheet Management, IT Service Management, Project Management, Change & Configuration Management, and Ad hoc Reporting.

Time Sheet Management

CENTRE provides a Time Sheet Management subsystem whose output can be directed to ODBC compliant financial management systems. At this time CENTRE Time Sheet data is formulated appropriately and exported to DELTEK for processing. For more information please see the “referenced” documentation.

IT Service Management

CENTRE’s IT Service Management subsystem provides a bidirectional data transfer capability with CA Unicenter, BMC Remedy and HP Peregrine. For more information please see our Bridge Server documentation

Project Management

CENTRE Projects, our Project Management subsystem provides an export/import facility for Microsoft Project. For more information please see the “referenced” documentation.

Change & Configuration Management

The CENTRE Change & Configuration management subsystem allows for the collection and recordation of data from such popular network management systems as Microsoft Essentials, or any other system that can perform ‘Discovery’, can tracks events and can forward identified records via email.

The ITG, patent pending and trademarked “Exchange Interface”, parses and converts into data base records the contents of any emails constructed in structured format. Where network management software is not implemented CENTRE can import Configuration Items from formatted Excel spreadsheets.

As well, CENTRE records may be exported to Microsoft applications such as MS Word and MS Excel.

CENTRE Interfaces Defined:

CENTRE – DELTEK Interface

CENTRE's Time Management subsystem is integrated within Project Management and very much dependant on data managed within Tasks. Because these modules are so intertwined, users are not able to record hours in their time sheets unless they are specifically assigned as resources in active tasks with valid work schedule period allocations. In order to be able to export time sheet data to the DELTEK system, a data element cross mapping is required between the CENTRE SQL table structure and the DELTEK file system. To do this, ITG developed a tool to map data elements between the two systems and creating an intermediate database where data transferred from the originating system is stored before transferred into the target system. See Appendix A for more detail on this Bridge Server application.

DELTEK tracks resource hours by maintaining a list of accounts, a list of employees, their rates, classification codes, and various other related information. Employee hours are recorded under a given account number and a DELTEK account number is equivalent to what is known as Contract Number in CENTRE. Since a project in CENTRE references a contract number, and the fact that a project has resources and these resources track their time in time sheets, we know that we are able to establish accurately table structure data element cross mapping between the two systems. After the data element cross mapping was generated via BridgeServer, what remains is a way to actually access the DELTEK file system. This is achieved with the purchase of an ODBC driver specifically developed for the DELTEK file system from a third party organization.

In addition to the above interface, a parsing application was developed to read a text file generated by accounting from DELTEK called Monthly Summary Report. This report includes financial performance data for each project that is tracked in CENTRE. The parsing application will automatically detect the presence of this file every month, and while parsing it, will import the monthly data and store it accordingly for each project within the CENTRE Project Financials module.

CENTRE – CA Service Desk Interface

CENTRE's IT Service Management subsystem is capable of processing service tickets that have been created initially in the CA Service Desk application. Using the ITG developed Bridge Server Tool, a temporary database has been configured to store CA Service Desk records that are retrieved via a stored procedure set up to execute against the remote database over an ODBC connection in preset time intervals (a configurable parameter). The data records, if any are retrieved, are then stored in the Bridge Server temporary database where further processing and data normalization takes place prior to insertion into the CENTRE service management related tables. The cross mapping mechanism provided by the Bridge Server tool makes this all possible to occur in real time. Data records can appear in CENTRE within seconds after being created in CA Service Desk. Furthermore, alerts have been built in to notify an administrator should a problem occur with the data connection between the two systems and/or during the data transfer. See Appendix A for more detail on the Bridge Server application.

CENTRE – Microsoft Project Interface

The CENTRE Project Management subsystem is a comprehensive module that enables a project manager to plan for a project with the use of tasks, resources, budgets, and estimates. It also provides the ability to monitor and track progress during the lifetime of the project. There is an abundance of data that is recorded, tracked, and maintained with the database table structures dedicated to the Project Management subsystem. An interface was developed within CENTRE to allow for the bidirectional data transfer between CENTRE and the Microsoft Projects application. The interface is web based and is accessible from within CENTRE. From a submenu option provided under the Project & Financial Management tab called “Import/Export”, a project manager with the proper authorization is able to export an entire project, including tasks and resources, to a Microsoft Project .MPP file. This new .MPP file can then be opened and edited using the Microsoft Project application. Similarly, a project manager is able, via the same interface, to import a Microsoft Project .mpp file (i.e. a project that was created using the Microsoft Project application), and create a project in CENTRE where the most critical information such as the project name, start and end dates of the project, all project tasks by name and with start and end dates, and assigned resources. The interface developed makes use of all relevant Microsoft APIs.

System Configuration

The first level of cross mapping is performed at the database level where the tables of the local database are mapped to the MS Project file system that supports project management.

[Required Columns](#) | [Export](#) | [Import](#) | [Back](#) | [Help](#) | [Sign Out](#) |

EXPORT IMPORT TABLE MAPPING		
MS PROJECT TABLE	DATABASE TABLE	
Project	Project	Delete
Resource	TaskMember	Delete
Task	Task	Delete
1		
*MS PROJECT TABLE	*DATABASE TABLE	
Select a ms project object	Select a database table	Add

The second level of cross mapping is performed at the column level where the desired or relevant data columns in MS Project are mapped to the corresponding local database table columns. The user interface developed to enable this functionality is illustrated below.

In the event that a direct mapping of a selected data column from the local database side is not possible and a second table needs to be referenced with a foreign key, the user interface provides the capability to configure this relationship. This relational specification is taken into account during the export/import of a project.

COLUMN MAPPING FOR EXPORT		
MS PROJECT COLUMN	DATABASE COLUMN	
MSPProject.ProjectFinish	Project.EndDate	Delete
MSPProject.ProjectStart	Project.StartDate	Delete
MSPProject.Title	Project.ProjectName	Delete
Resource.Name	Users.UserName	Delete
Task.ActualFinish	Task.ActualEndDate	Delete
Task.ActualStart	Task.ActualStartDate	Delete
Task.Finish	Task.EndDate	Delete
Task.Name	Task.TaskName	Delete
Task.Start	Task.StartDate	Delete

1

MS Project Object:

MS Project Column:

Database Column:

Primary Key Table:

Primary Key Column:

Foreign Key Table:

Filter:

Foreign Key Column:

Foreign Key Column containing Value:

In addition, the application provides a user interface to allow the user to specify a required set of data columns that have to be populated with data in order for the export or import function to execute. This user interface is illustrated below.

Once again, in the event that a direct mapping of a selected required data column from the local database side is not possible and a second table needs to be referenced with a foreign key, the user interface provides the capability to configure this relationship. This relational specification is taken into account during the export/import of a project.

REQUIRED COLUMNS					
TABLE NAME	COLUMN NAME	COLUMN VALUE	DATA TYPE	MAXIMUM LENGTH	
Project	LastModDate	getdate()	datetime	8	Delete
Project	Status	19	int	4	Delete
Task	SiteID	38	int	4	Delete
Task	StartDate	getdate()	datetime	8	Delete
Task	EndDate	getdate()	datetime	8	Delete

1

Primary Key Table:

Column Value:

Data Type:

*Max Length:

Primary Key Table Column Name:

Foreign key table:

Foreign key column:

Filter:

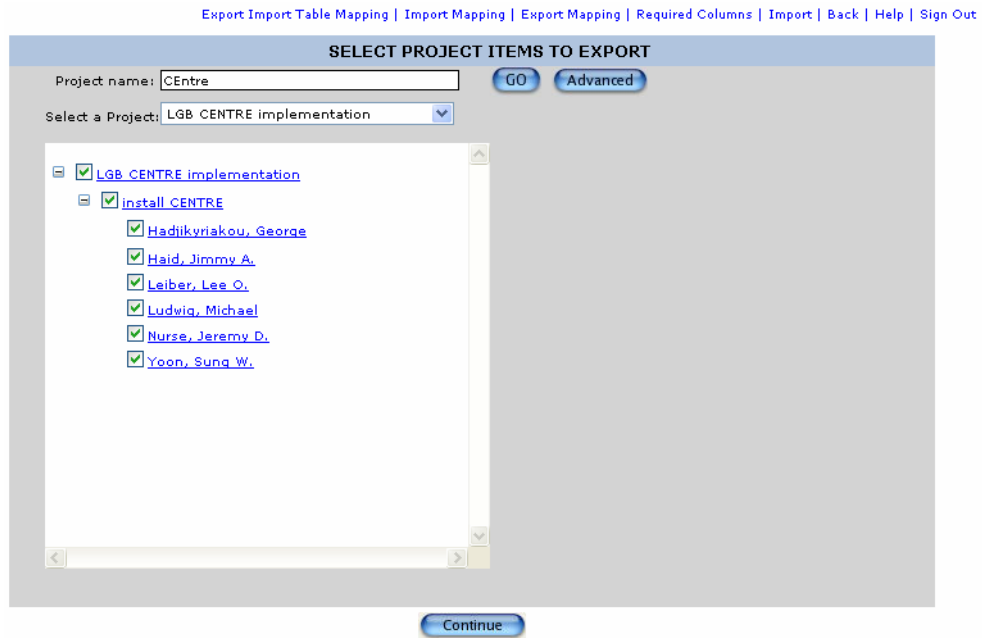
Foreign key table column containing Value:

column Value:

Exporting/Importing

A simple but effective user interface is provided where a user will be able to search for and select a project from the local database to be exported to an MS Project file. The exported project will be in the form of .MPP file, MS Project native file format. This new file will be capable of being opened and edited using the MS Project application.

The user interface developed to begin the export process is illustrated below. The user can provide a partial string of the name of the desired project to export and select the GO button. This action will display one or more projects that were found matching the partial string provided and displayed in the tree-view text box. If the user wishes to use additional filter parameters to locate a project, the Advanced search button is provided and when selected, a more detailed search screen is displayed enabling the user to use other parameters in the search query.



The selected project will appear in the tree-view and the user has the option of not only selecting the entire project but also to deselect any of the tasks or resources of a task so that they are excluded from the export process.

Once the selection is made, selecting the Continue button will display a list of all of the tasks and resources under each task as confirmation. The user will be able to also select or deselect various items before the export is performed.

CONFIRM PROJECT ITEMS TO EXPORT		
<input type="checkbox"/>	PROJECT NAME	New Project
<input type="checkbox"/>	LGB CENTRE implementation	New Project
<input type="checkbox"/>	LGB CENTRE implementation	New Project
<input checked="" type="checkbox"/>	LGB CENTRE implementation	New Project
TASK NAME	New Task	RESOURCE(S)
<input checked="" type="checkbox"/>	install CENTRE	New Task
		<input type="checkbox"/>
		<input checked="" type="checkbox"/> Hadjikyriakou, George New User
		<input checked="" type="checkbox"/> Haid, Jimmy A. New User
		<input checked="" type="checkbox"/> Leiber, Lee O. New User
		<input checked="" type="checkbox"/> Ludwig, Michael New User
		<input checked="" type="checkbox"/> Nurse, Jeremy D. New User
		<input checked="" type="checkbox"/> Yoon, Sung W. New User

When the export takes place and completes, the application will report a summary of the number of tasks and the number of resources that were exported. This serves as a final confirmation.

A similar set of user interfaces are available to enable the import of a project from MS Project to the local database

CENTRE – Microsoft Exchange Interface

The CENTRE email detection and parsing interface application enables CENTRE to construct multiple email detection and parsing schemes along with specified email accounts to be monitored. An interface was developed within CENTRE to allow users to construct email detection and parsing schemes used to identify and log emails for relevant content from the specified email account inbox.

The CENTRE Microsoft Exchange Interface application enables the user to create one or more email detection and parsing configuration modules. In each of these modules, the user can specify 1) a related database table, 2) the IP address of the Microsoft Exchange server, 3) the email account address that will be monitored along with that email account’s login credentials, 4) the frequency on which the applications will be checking for new email arrivals, 5) and whether the data in the emails identified using this module configuration will be used to create records in the specified database table. One major objective of this implemented solution is to provide the ability to assemble and insert a new record in any of the database tables by using the data contained in an email sent and received specifically for this purpose.

MODIFY MODULE PARAMETERS	
*Module Name:	Incident - Service Record
Table Name:	SrvCall
Email Account:	demo.demo@itgdemo.com
MS Exchange Server Name:	172.16.10.124
*User Name:	demo.demo
Password:	Logmeint
Check Frequency:	1
Create Record:	<input checked="" type="checkbox"/>

Part of the configuration capability provided enables the user to identify data elements expected to be found in the subject line of the emails processed by this detection and parsing module

configuration. As is normally the case with relational databases, the string value detected in the subject line may not be the actual value saved in the record, should this configuration require that a record be created from an email received. Therefore, the user interface illustrated below enables the user to not only identify the data element expected to be found in the subject line of the emails destined for processing by this configuration module, but it also enables the user to identify a foreign key relationship that will be used to locate the value to be used during the new record creation.

Similar to the subject line configuration, the application provides a user interface to specify data elements expected to be found in the main body of the emails processed by this detection and parsing module configuration. This is the data that will be used in the event that the configuration dictates that a new record will be created from the data received. Once again, should some of the data elements specified require the identification of foreign key relationships, the application user interface developed will facilitate this, as illustrated in the screen image below.

A background process was developed as part of this solution, to constantly monitor the specified exchange email account inboxes at a set frequency for emails of interest. If any emails meeting the configured criteria are detected, they are logged, and if any require an action to be carried, the system will do so and capture the details in a log that can be viewed by the user at a later time.

CAPTURED MODULE RECORDS						
March		April 2009			May	
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Enter Duration To View Log
 From :
 To :

MODULE NAME	KEY COLUMN	RECORDID	SOURCE EMAIL	DATE CREATED
Incident - Service Record	SrvcCallID	143074	demo demo	04/13/2009
Incident - Service Record	SrvcCallID	143076	demo demo	04/13/2009
Incident - Service Record	SrvcCallID	143078	demo demo	04/13/2009
Incident - Service Record	SrvcCallID	143082	demo demo	04/13/2009
Incident - Service Record	SrvcCallID	143084	demo demo	04/13/2009

The interface is web based and is accessible from within CENTRE. From a submenu option provided under the System Administration tab, a user with the proper authorization is able to check the email log which is constantly being updated to indicate outgoing and incoming emails. The interface developed makes use of all relevant Microsoft Exchange APIs.

EMAIL LOG SHOWS THE DETAILS OF THE RECORDS CAPTURED						
March		April 2009			May	
Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Enter Duration To View Log
 From :
 To :

MODULE NAME	KEY COLUMN	RECORDID	SOURCE EMAIL	SUBJECT	DATE CREATED	BODY	Statuscode
Incident - Service Record	SrvcCallID	143086	demo demo	Incident - Service Record	04/14/2009	ContractID: 44 ServiceRecordCategory: CallRefNum: 137699 RequestRefNum: EventRefNum: CustRefNum: CustOrderNum: CustPOC: George Hadjikyriakou CustPOCPhone: 703-698-8282 CustPOCPhoneExt: 259 CustPOCEmail: georgeh@itgonline.com AltPOC: George Hadjikyriakou AltPOCPhone: 703-698-8282 AltPOCPhoneExt: 259 CallerName: George Hadjikyriakou CallerPhone: 703-698-8282 CallerPhoneExt: 259 Agency: ITG Addr1: 2745 Hardland Rd. City: Falls Church State: VA ZipCode: 22043 MfrName: OTHER Model: 4100 SerialNumber: ITG13860700 Description: Printer AssetTag: 00000	Ok

CENTRE – Microsoft Essentials Interface

In the Asset Management section of the CENTRE Contract Risk & SLA Management module, an application was developed to parse through the Microsoft Essentials generated device discovery Excel file that contains device data

CENTRE – Microsoft Excel and Microsoft Word Interface

In the CENTRE Ad Hoc Reports module, a function was added to allow the user to export the resulting data from a report to Microsoft Excel or to Microsoft Word.

In the Asset Management section of the CENTRE Contract Risk & SLA Management module, an application was developed to import records provided in a Microsoft Excel file.

APPENDIX A

CENTRE Import/Export Tool - Bridge Server™.

BridgeServer is a data extraction tool used to organize the exchange of information between databases. *BridgeServer* provides a user interface to the table structure of target databases and facilitates the identification of areas containing similar kinds of information within two database systems.

Many databases contain hundreds or even thousands of fields and tables. Only a small portion of the information contained within a database may be of interest for the task at hand. *BridgeServer* allows users to define *Retrieval Schema* to select only those portions of each database to read during periodic extractions. Further, *Date Filters* are used to restrict the extracted data to a particular time period. Any date field within a database may be used to filter the data pulled during the extraction. These parameters reduce the time spent polling target databases and lessen the extraction of irrelevant data.

BridgeServer allows disparate databases to exchange information through its use of *Cross References*. *Cross References* are user specified fields containing identifying information. Ideally, target databases already contain fields created specifically for identification purposes (i.e. social security numbers, driver's license, serial number etc). Quite often, however, such fields are not available in both databases. Moreover, using these fields for tracking purposes is increasingly forbidden by legislation designed to protect privacy (i.e. HIPAA). *BridgeServer* cross-references may span several fields and *normalization functions* may be used to extend matching efficiency. Matches may also be defined manually. *Cross-Referencing* allows *BridgeServer* to track records in the absence of unique identifiers. Records that remain unmatched after a data exchange are easily identified.

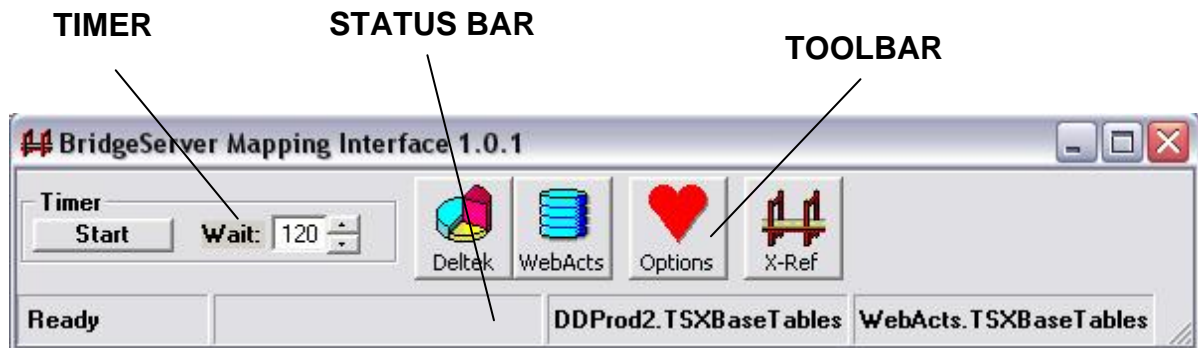
BridgeServer is intended to be used either as middleware – it can supply primary applications with data extracted from one or more database systems – or as a development tool where it can simplify apparent database structure and allow developers to isolate tables and fields containing data of interest.

BridgeServer is read-only in its interaction with target databases. *BridgeServer* cannot be used to delete, update or insert records within its target databases, nor can it be used to modify target tables, fields or indices. *BridgeServer* is discreet and unobtrusive.

BridgeServer maintains internal copies of each data exchange. Thus, *BridgeServer* can be used to audit database changes over time. Users may configure alerts to be sent to specific email addresses in the event of a scan error. *BridgeServer* keeps an internal log of all errors and their causes.

The *BridgeServer* MDB (Microsoft Access Database) does not require an ODBC DSN (Open DataBase Connectivity Data Source Name). However, a system ODBC DSN must be configured for any database that *BridgeServer* will use as a target.

The Bridge Server User Interface



BridgeServer's Main Screen

1	TIMER TOOL		Timed Extractions
2	MANUAL EXTRACTION		Manually Scan Target Databases
3	OPTIONS		Setup Target Databases, Retrieval Schema and <i>BridgeServer</i> Options
4	CROSS REFERENCES		Setup Cross References
5	STATUS		Program and Extraction Status DSN Definitions and Retrieval Schema

In the *BridgeServer* application shown above, the target databases are the Deltek Accounting System and ITG's WebActs (ITG-Centre).

1. **The Timer Tool** is used to schedule periodic extractions. Use the Wait Box to setup the idle time (in minutes) between scans. The "Start" button initiates the extraction cycle. For the screen above the cycle is as follows:
 1. The Deltek DSN is extracted according to TSXBaseTables (The Deltek DSN current Retrieval Scheme)
 2. WebActs DSN is extracted according to its TSXBaseTables (The current retrieval scheme for WebActs)
 3. The system is idle for 120 minutes.
2. **Manual Extraction Buttons.** Pressing a Manual Extraction Button starts a scan of the target using the Retrieval Scheme shown in the Status Bar. The subtitle of each manual extraction button displays the name of the target database DSN.

3. The **Options Button** provides access to screens where the user selects the target datasources, defines retrieval schema, and configures alerts.
4. The **Cross Reference Button** is used to configure data cross-references.
5. The **Status Bar** shows the names of the currently configured target DSN's and their selected retrieval schemes. When a scan is in progress, the **Status Bar** shows the current operation.

Configuration Options

Click the Options Button  to access the Options Screen.

BridgeServer's Options Screen

1	SETTINGS	
2	ERROR HANDLING	

3	ALERTS	ALERTS <input type="checkbox"/> Enable Alerts SMTP: <input type="text"/> PPM: <input type="text"/> TREREP: <input type="text"/> OFF-PPM: <input type="text"/> WindoPath: <input type="text"/> ADMIN: <input type="text"/>						
4	PPM	PPM Start: Hour: <input type="text" value="8"/> Min: <input type="text" value="0"/> End: Hour: <input type="text" value="17"/> Min: <input type="text" value="0"/> <input checked="" type="checkbox"/> Closed Saturday <input checked="" type="checkbox"/> Closed Sunday HOLIDAYS <table border="1"> <thead> <tr> <th>HolidayID</th> <th>Date</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="height: 50px;"> </td> </tr> </tbody> </table>	HolidayID	Date	Name			
HolidayID	Date	Name						
5	DSN Mapping	DDProd2 DSN: <input type="text"/> DATABASE: <input type="text" value="Deltek"/> USERNAME: <input type="text"/> PASSWORD: <input type="text"/> <input type="button" value="Mapping ..."/> WebActs DSN: <input type="text"/> DATABASE: <input type="text" value="WebActs"/> USERNAME: <input type="text"/> PASSWORD: <input type="text"/> <input type="button" value="Mapping ..."/>						

The Options Screen is divided into five Sections.

The **Settings Section** controls the About Box. Check **Show Splash at Startup** if you want *BridgeServer* to announce itself whenever it is started. Click **“About”** to show *BridgeServer*’s About Box.



***BridgeServer* About/Splash Screen**

The About Box has a button to display System Information. The “ITG Online” button is a link to ITG’s website.

The **Error Handling** Section allows the user to decide which action to take in the event of a system or scanning error. The options are “Never Terminate” or “Terminate After X Errors” where X is the maximum number of errors to tolerate. Scanning errors may be caused by a variety of factors, including network accessibility of target databases and SQL errors caused by the retrieved data itself. *BridgeServer* maintains an internal log of system and scan errors. The Error Log displayed when the “Error Log” button is clicked.

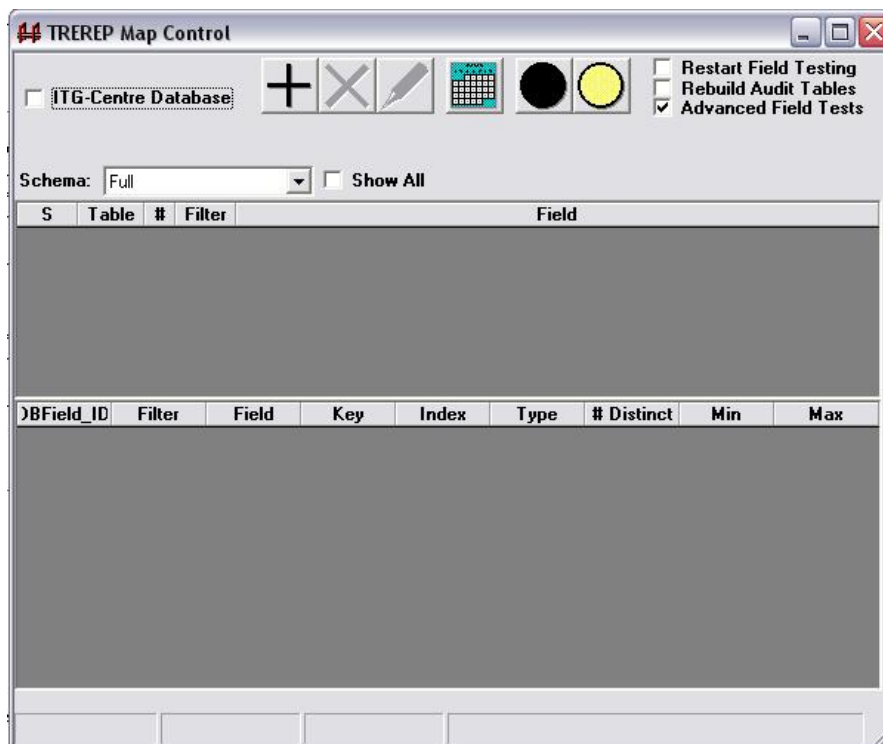
In the **Alerts Section**, check “Enable Alerts” if you would like an alert to be sent in the event of abnormal program termination or error.

Use the **SMTP Box** to specify the out-going (SMTP) mail server. You may specify up to five email addresses that will receive email alerts:

Use the **DSN Mapping Sections** to define two target System DSNs.

1. Select a defined DSN from the **DSN Dropdown List**.
2. Name the database (it is usually best to give the database the same name as the DSN) using the **Database Textbox**.
3. Enter the Username and Password required for the database (if required).
4. Click the **Mapping Button** to bring up the DSN Mapping Screen.

DSN Mapping, Retrieval Schema and Filters



BridgeServer's Map Control – Before Mapping

To map a Target DSN

1. Check the **ITG-Centre Database** checkbox if you are mapping an ITG-Centre database.
2. Select Mapping Options
 - a. **Restart Field Testing** – select this option if you want to restart the mapping process for the DSN from the beginning. Occasionally databases will have fields

that cannot be mapped. Attempting to access these fields may cause a system error (blue screen). These fields are left in the distributed database either accidentally or to prevent programmatic entry.

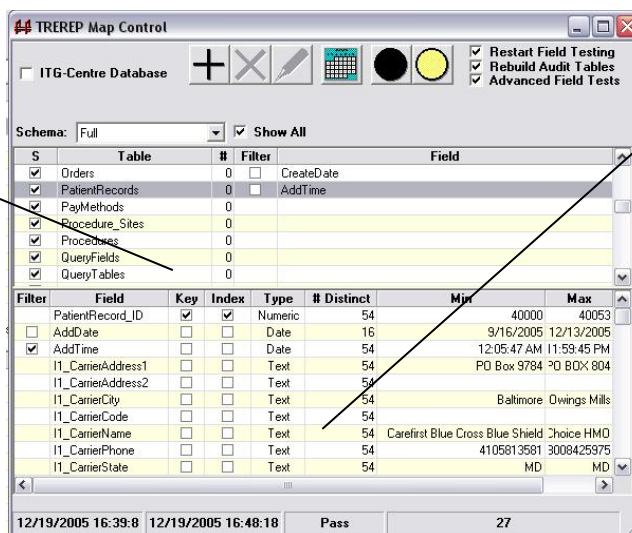
BridgeServer is able to map databases containing these booby- trapped fields by remembering which fields cause memory access problems. Check the **Restart Field Testing** checkbox if you want to completely remap the target DSN. If you have encountered an error during the mapping process and you want *BridgeServer* to start mapping from the last good entry point, leave this box unchecked.

- b. **Rebuild Audit Tables** – Check this checkbox if you want *BridgeServer* to restart the Audit trail for the DSN being mapped.
- c. **Advanced Field Tests** – If you want *BridgeServer* to record each field’s maximum value, minimum value and distinct record count during the mapping process check this box.

- 3. Click **Map Database**  Button to start mapping.

**MAP CONTROL
FIELDS GRID**

**MAP
CONTROL
TABLES
GRID**



BridgeServer’s Map Control – Tables and Fields Grids







During the mapping process the **Status Bar** displays the time at which mapping started and the time of the most recent mapping operation. The name of the table and field being currently undergoing the mapping procedure is also displayed. Please note that some tables and fields may require a long time to finish mapping, especially if **Advanced Field Tests** are selected.

When mapping is completed the **Mapping Control** displays the Tables and Fields available for the selected DSN. The Tables and Fields grids of the mapped DSN are shown in the figure above. From left to right the Tables Grid lists whether the table is selected in the current Retrieval Scheme, the name of the table, the number of records in the table, and the status of any selected data filter fields. *Click the row of the Table Grid to show the fields within a particular table in the Fields Grid.* The Fields Grid lists (from left to right) whether the field may be used as

a date filter, the name of the field, whether the field is a primary key or otherwise indexed, the data type of the field, the number of unique values contained by the field, and the minimum and maximum values of the field. This information is often extremely valuable to software developers.

The mapping process automatically creates the first Retrieval Scheme for the target database. The default Retrieval Scheme is entitled “Full” and contains all tables and fields in the target database. If an extraction is performed while the selected Retrieval Scheme is “Full” each table and field in the database will be read and audited. Depending on the size of the database target, this may take a very long time.


The **Mapping Control Toolbar** is used to define Retrieval Schema, Date Filters and to control Audit Processes.

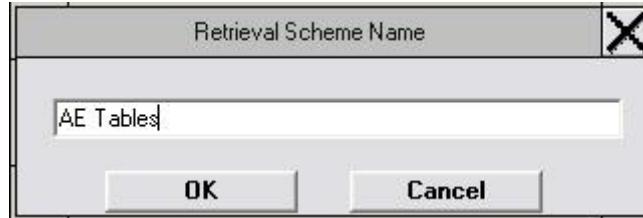
1		Add Retrieval Schema
2		Delete Retrieval Schema
3		Name Retrieval Schema
4		Define Data Filters
5		Clear Audit
6		Map DSN

Using the Retrieval Scheme Control Screen:

- Use the **Schema Dropdown** to select the current Retrieval Scheme
- Use the **Show All** checkbox to make all the tables appear in the DSN Table Grid. If the **Show All** checkbox is unchecked, only the tables selected for the current Retrieval Scheme will be displayed
- Check or uncheck the “S” column of the tables you want to include or exclude from the selected Retrieval Scheme.

To Add a New Retrieval Scheme:


1. Click the **Add Retrieval Scheme Button** .
2. Enter the name of the new scheme in the **Retrieval Scheme Name Pop Up**:



3. Select the Tables you would like to include in the Scheme by clicking the checkbox in the “S” column of the Tables Grid.


To Delete a Retrieval Scheme:

Select the Retrieval Scheme you want to delete in the **Schema Dropdown**.

1. Click the **Delete Retrieval Scheme button** .
2. Confirm the Deletion in the **Deletion Pop Up**:




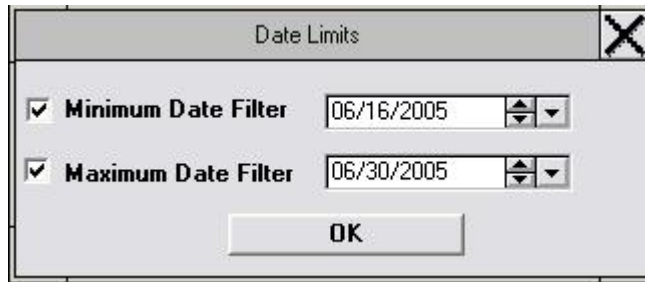
To Rename a Retrieval Scheme:

1. Select the Retrieval Scheme you want to rename in the **Schema Dropdown**.
2. Click the **Rename Retrieval Scheme button** .
3. Enter the new name in the **Retrieval Scheme Name Pop Up**:



To Add a Date Filter to a Retrieval Scheme:

1. Select the Retrieval Scheme you want to rename in the **Schema Dropdown**.
2. Click the **Set Date Filters Button** .
3. Select the limiting dates in the **Date Limits Pop Up**:




If you want to specify the earliest allowed date in the extracted data check the **Minimum Date Filter Box**. If you want to specify the latest allowed date in the extracted data make sure the **Maximum Date Filter Box is checked**. Thus, checking both the Maximum Date Filter and Minimum Date Filter Boxes will return the inclusive range of dates for the records in the selected table.

To reset all of the Audit Data retained for a Target DSN:

Click the **Audit Reset**  **Button**.


Cross Referencing

Click the **Cross Reference Button**  to access the **Cross Reference Control Screen**.

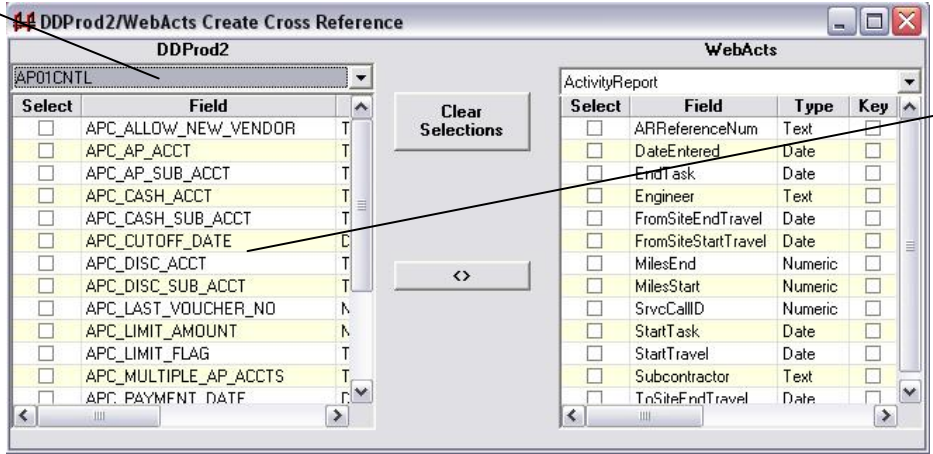


BridgeServer's Cross-Reference Control Screen

To Add a New Cross Reference:

1. Click the **Add Retrieval Scheme Button** .
2. Use the **Table Dropdown Boxes** and the "Select" column of the **Fields Grid** to select the fields that will comprise the Cross Reference for both DSNs.

**XREF CONTROL
Table Dropdown**



**XREF CONTROL
FIELDS GRID**

BridgeServer's Create Cross-Reference Screen

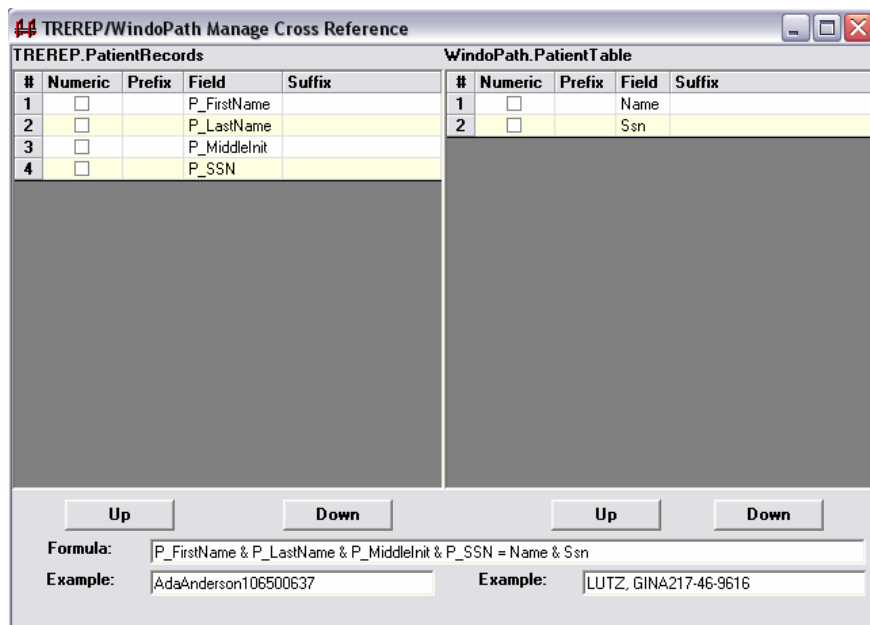
3. Click the **Create Cross Reference**  **Button** to create the Cross Reference and proceed to the **Cross-Reference Management Screen**.

To Delete a Cross Reference:

1. Select a cross-reference from the **XRef Grid** of the **Cross-Reference Control Screen**.
2. Click the **Delete Cross Reference**  **Button**

To Manage a Cross Reference Definition:

1. Select a cross reference from the **XRef Grid** of the **Cross-Reference Control Screen**.
2. Click the **Cross-Reference Management**  **Button**.



BridgeServer's Cross-Reference Management Screen

Use the **Cross-Reference Management Screen** to fully define the Cross Reference. The fields selected in the **Create Cross Reference Screen** are shown for both DSNs.

*A Cross-Reference is made of two string formulae.
A Cross-Reference Match occurs when
 $F1(DSN1) = F2(DSN2)$*

Where F1 and F2 are the string formula and DSN1 and DSN2 are the field values from the target databases.

String formulae consists of the following elements:

1. **Sampling Order** - The order in which each field is taken. Select a row in the **Fields Grid** and use the **Up** and **Down Buttons** to change the sampling order.
2. **Normalization Functions** – checking the Numeric checkbox for a field will force the use of only the numeric characters for the field. Punctuation and alphabetic values will be ignored for matching purposes.

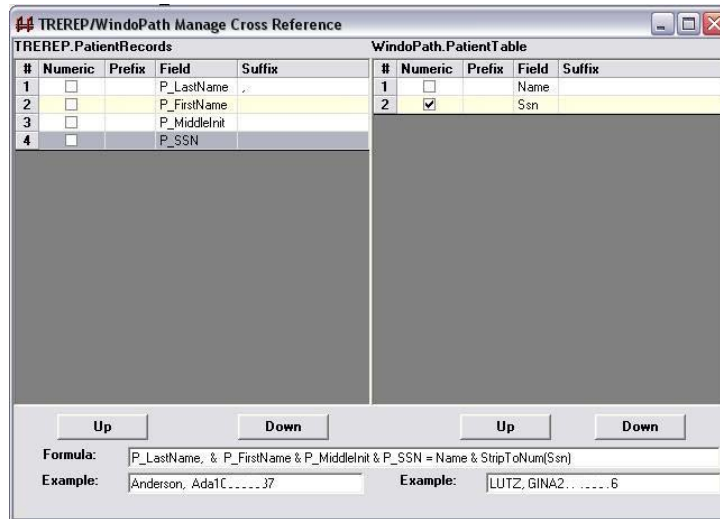
Thus, the numeric normalization function acting on a phone number field will be interpreted as '3015551212' instead of the actual value (301) 555-1212. Similarly, Social Security Numbers will be seen as 9 digits by the formula, even if the source database allows formatting hyphens.

Other normalization functions may be applied behind the scenes, for example all date fields are converted to a specific format when used as part of a cross-reference for matching purposes.

3. **Prefix and Suffix entries** – matching may require string appendages to be added before and/or after field values respectively. In the above example a comma and a space must be placed between the last and first name fields retrieved from DSN1 in order to achieve a match.

The equating string formulae are shown in the **Formula Textbox**. Results of the application of the formulae applied to example fields from each DSN are shown in the two **Examples Textboxes**.



The following screen shows how the cross-reference defined in this example is finally formatted to create matches in the extractions of the two DSNs.



1. The Name field in DSN2 is a person's last name, first name and middle initial combined into a single field. The Sampling Order of three fields in DSN1 was manipulated to achieve a match.
2. Hyphens in the Social Security Number of DSN2 were removed by checking the Numeric Text box.

The SSN numbers appearing in the Example Textboxes have been purposely obscured in this example. Please note that it is not necessary for the values shown in the Example Textboxes to match. The Examples simply show the application of the string formulae to example fields of each DSN. Also, the above example - shown only for illustrative purposes - would obviously have been simplified by using just the SSN field that is available in both databases. The Numeric normalization function applied to DSN2 would suffice for the match.

To Test a Cross Reference:

1. Select a cross-reference from the **XRef Grid** of the **Cross-Reference Control Screen**.
2. Click the **Run Cross Reference Table**  **Button** on the **Cross Reference Control Screen**.
3. Click the **Run**  **Button**.

The number of unique cross-references is shown under the Cross Reference table. Records that cannot be cross-referenced from each DSN appear in the bottom half of the screen.

Orphan Records and Manual Cross References

The above examples show that unmatched field records remain even after the application of the string formulae. Unmatched records may signify that the companion database simply has no corresponding record. Quite often, however, orphan records are caused by data entry procedures. In the second example above, employees' names cannot be matched because of the use of nicknames, inconsistent use of middle initials, or simple spelling mistakes during data entry.

BridgeServer allows the user to match orphan records manually. Once orphan records are manually linked, they will remain linked in future extractions. Obviously, manual matching is difficult if there are a large number of orphan records.


To Create a Manual Cross-Reference

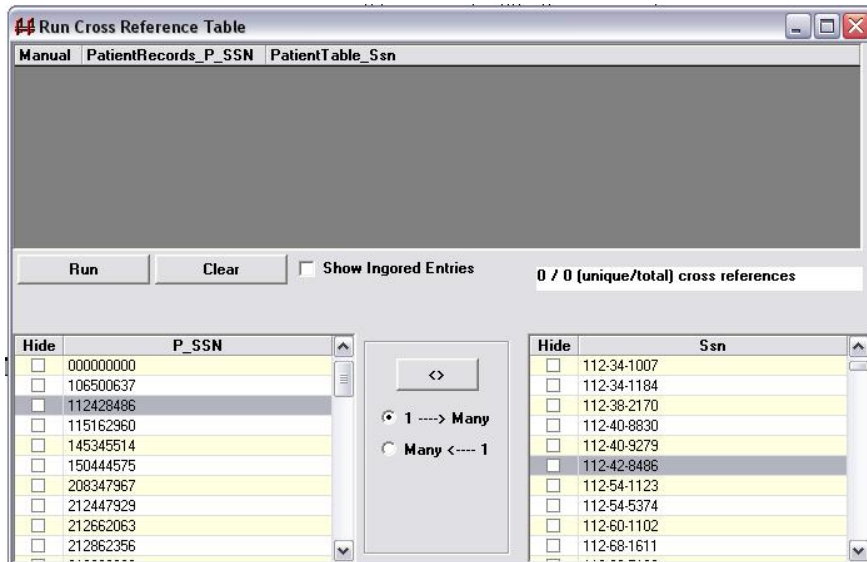
1. Choose the direction of the cross-reference. A single entry from DSN1 may match many entries from DSN2. Once matched, the entries need to be removed from the Manual Cross-Referencing Grids.

Select the direction of the manual cross-reference by choosing one of the options:



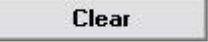
Choose the top option if a record from DSN1 may match many records from DSN2. Choose the bottom option if a single record from DSN2 may match many records in DSN1.

2. Use the Hide checkbox column and the Show Ignored Entries checkbox as needed to facilitate manual matching. These checkboxes are used to facilitate the matching process by toggling the visibility of records for which there is no match.
3. Once a match has been identified, click the corresponding rows of each grid and click the manual match  button.
4. Records that have been manually cross-referenced appear with a check mark in the Manual Column of the top grid (see example 2, above).



BridgeServer selecting records for manual matching

To Clear (Restart) a Cross Reference:

Click the **Clear**  **Button**. All cross-references will be removed and all records will be available for manual cross-referencing.

BridgeServer.MDB

All results are contained in BridgeServer.MDB. This file is located in the *BridgeServer* installation directory “C:\Program Files\BridgeServer”. It is a Microsoft Access Database.

Two copies of each extracted table are kept in BridgeServer.MDB. The tables are named after their originating database and table with either *_Audit* or *_Copy* appended to their names.

_Audit tables contain the most recently extracted data. *_Copy* tables contain extracted information and date/time stamps noting record additions, record deletions and time of last update. Although these names seem backwards now, it made sense to name them this way during development.

The *_Audit* and *_Copy* tables of BridgeServer.MDB have separate primary keys. BridgeServer.MDB keys are used to maintain audit records and to provide a single unique index to programmers who may otherwise have to deal with multiple key fields and other antiquated database techniques. Primary keys of source databases are used only to prevent duplicate records in *BridgeServer*.

Each record in an *_Audit* table contains a foreign key reference its corresponding *_Copy* table master record. With this key it can be determined when a particular record was originally added and when it was last updated.

Each *_Copy* table is supplemented with the following fields:

DSN_TableName_Copy_ID	<i>BridgeServer</i> Primary Key
DSN_TableName_Audit_ID	<i>BridgeServer</i> Foreign Key to <i>_Audit</i> Table
BS_IsFound	Used Internally during Audit
BS_IsNew	Used Internally during Audit
BS_IsDeleted	Used Internally during Audit
BS_AddDate	Date First Extracted
BS_DeleteDate	Date First Missing
BS_UpdateDate	Date Last Updated
BS_IsHidden	Used Internally during Manual Cross Reference
BS_IsManual	Manual Internally during Manual Cross Reference

Other tables in BridgeServer.MDB are:

- **BmailAlerts** is a list of all alert recipients and their roles.

- **DBTables** and **DBFields** list the tables and fields that have been mapped by *BridgeServer*, respectively. These tables also contain information pertaining to source database key fields and sample data.
- The **DBUpdates** table is a list of updates to any field in an extraction table. It lists the DSN name, table name, field original value of the field, the updated field value and the date of the update.
- **FieldTest** includes the test results from each field if the “Advanced Field Test” Option was selected during mapping. These results include whether the field is used by the source database as a primary key or index, the field data type, whether the field passes integrity tests, the number of distinct values the field contains, its maximum and minimum values (during mapping), and any internal or system errors accessing the field during mapping may have caused.
- **RetrievalSchema** and **RetrievalSchema_DBTables** are used internally by *BridgeServer* to track Retrieval Scheme definitions.
- The **Scans** table tracks each extraction by target database, date, duration, number of new records, deleted records and updated records and total records extracted.
- **SystemErrors** tracks each error encountered during a scan. Errors are usually caused by network accessibility problems or by error conditions existing in the data.
- The **UserOptions** table tracks data used by *BridgeServer* internally.
- The **Xref** and **Xreffields** tables are used internally by *BridgeServer* to define Cross References.
- **Xref_** tables (**Xref_TableName1_TableName2**) contain matching cross-references by *BridgeServer* Primary Key value.