



**Connex White Paper
Version 2.1**

Document Control	
Author	Johan Pienaar
Version	2.1
Date	26/09/2006

© Peresys (Pty) Ltd 2006. The copyright in this work is vested in Peresys (Pty) Ltd. The contents hereof are private and confidential. This document may not be reproduced in whole or in part except with the written consent of Peresys and then on condition that this notice is included in such reproduction. No information as to the contents or subject matter of this document may be communicated, whether oral or written, to any other party without prior written consent.

Table of Contents

Overview	3
Connex Features	4
Architecture	5
Connex Implementation	11
Future Releases	12

Overview

Connex is a middleware product, purpose built to perform rule-based data extraction, formatting, routing and other user defined tasks to achieve application integration. Connex provides customers with the ability to:

- Integrate disparate systems
- Connect to various data sources
- Automate business processes
- Automate the generation and processing of business messages
- Create and output data in a variety of formats
- Interact with databases and message queues
- Communicate with e-mail clients
- Utilize a simple drag-drop-interface with components as building blocks
- Construct transformation procedures
- Schedule transformation processes
- Facilitate controlled user interaction through the Message Viewer product

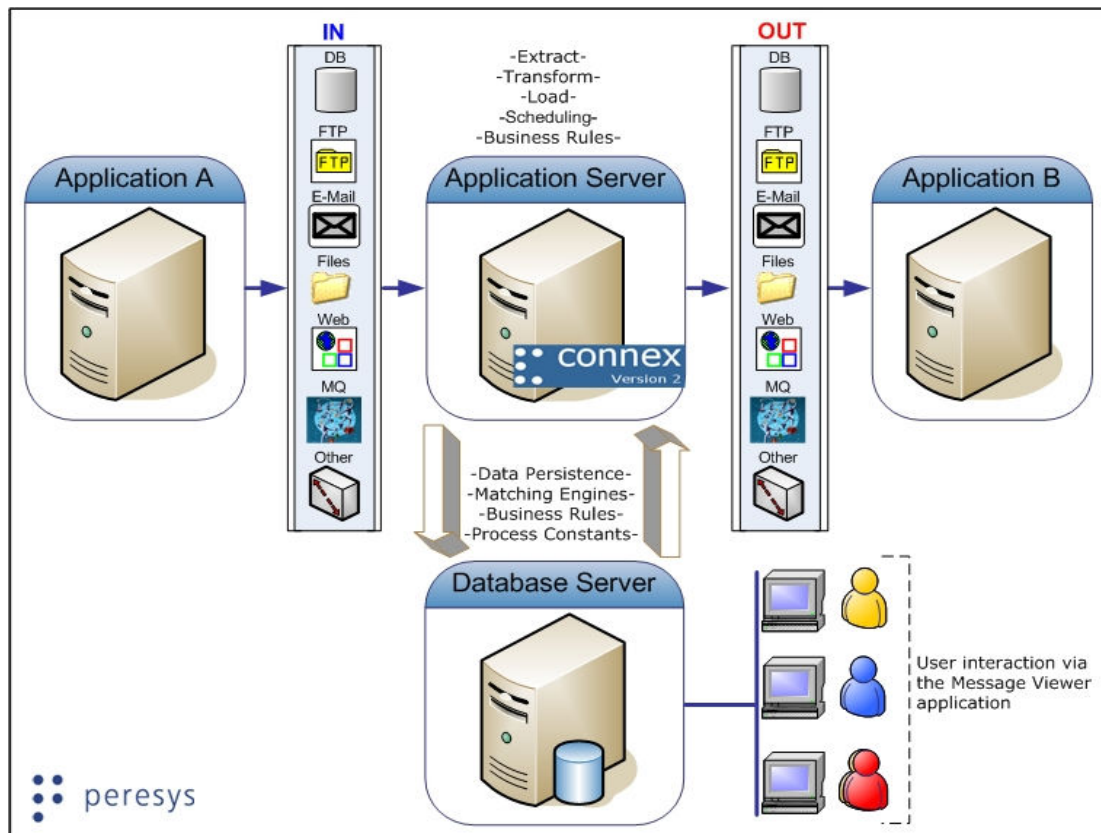


Fig1: Middleware traditionally resides between disparate systems with their own sets of inputs and outputs. It connects these systems through applying various business rules, which include validation, translation and transformation of data.

Connex has an intuitive, easy to use interface. Users do not need to learn any proprietary scripting language to define transformations nor do they require in-depth programming skills. With limited training, users with solid computer skills can be productive within a relatively short period of time. The training also covers best practices and methodologies which is a culmination of Peresys' integration experience in the financial services arena.

Connex Features

- **Ease of use** – It enables users who understand the business requirements to define and implement their own processes to transform, route, modify and / or enhance data without having to resort to programming.
- **Existing components** – Connex has an extensive set of off-the-shelf components to satisfy most of the integration requirements in typical corporate back-office environments. These components provide seamless integration to specific application software packages, thus further speeding up implementation.
- **Database Support** – Database connectivity is provided via ADO and ODBC. This caters for integration solutions where data persistence is required. Components for interaction with tables, views and stored procedures are part of the core functionality.
- **Speed** – Connex uses early binding on its components which ensures superior performance.
- **Multi-threading** – Server Admin, the Connex orchestration engine, is completely multi-threaded therefore allowing several tasks/procedures to be executed concurrently.
- **Debugging** – Transformations and processes can be fully debugged within the designer before being implemented.
- **Error handling** – Each Connex procedure contains an error branch, which can be configured to appropriately deal with any errors as they occur. Error information is also automatically written to the Connex log file.
- **Security** – A user management module is embedded with each Connex solution. This provides an interface where different users can be configured per solution, each with specific design and execution privileges.
- **Transactional support** – Wherever possible, Connex provides the framework for implementing transactional processes where roll back of incomplete transactions is required.
- **Extensibility** – The components used by Connex support a specific COM interface that can be utilized by the licensee to write proprietary components, as and when required. These components could loosely be referred to as software connectors.

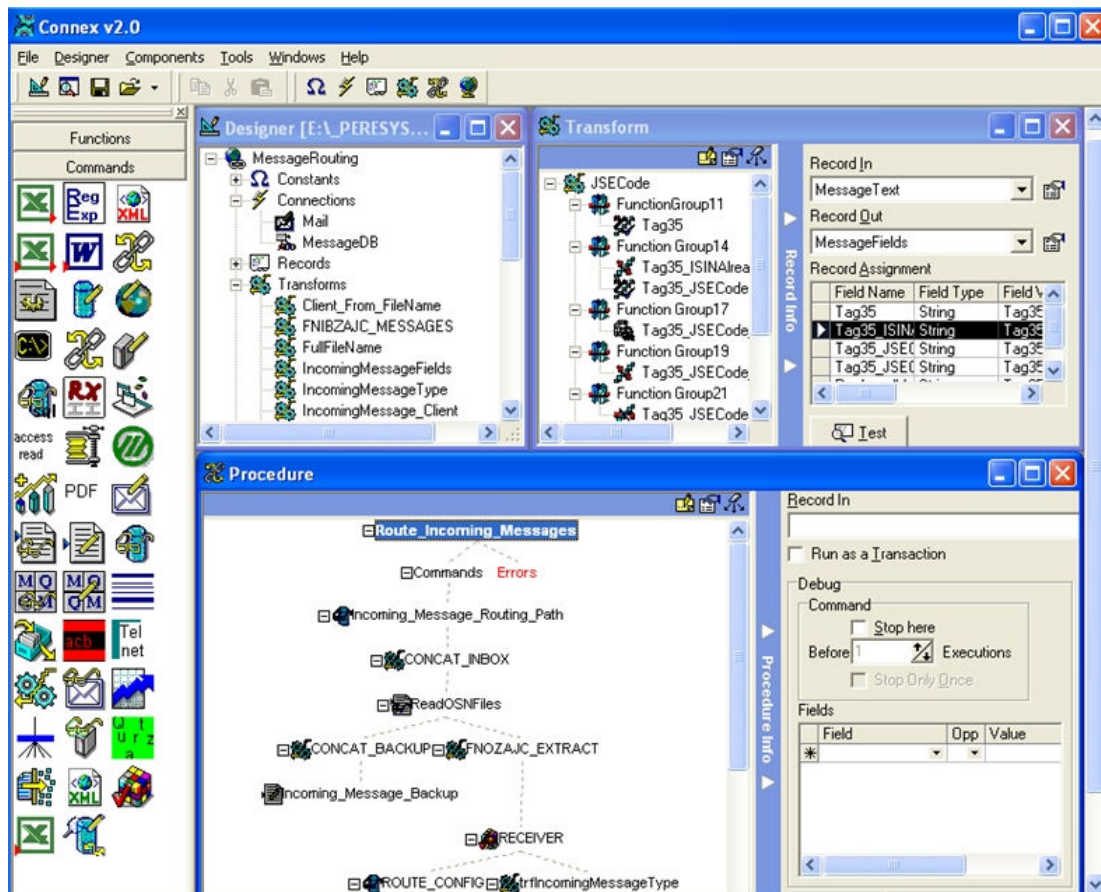
Architecture

Connex consists of three distinct modules:

- **Designer**
- **Server Admin**
- **Remote Admin**

Designer

The Connex Designer provides an easy to use drag and drop interface where the user can create integration solutions:



Using the designer, connections to databases, external systems and mail are set up together with data definitions, transforms, procedures and schedules to facilitate / automate the data transformation process. Procedures are designed and visually represented in a tree-like structure, where any number of branches can be defined. All information pertaining to any one solution can be stored in a single Connex **solution** (.cnx) file, which allows for simplified migration and backup procedures.

Functionality is also provided to test each transformation object and debug each procedure tree within the designer.

Each Connex component encapsulates the functionality required to support a particular business transformation. The components are grouped in libraries installed with the Connex suite.

The components are based on Microsoft COM technology and use a standard pre-defined interface (API) for communication with the designer and the server. For this reason components can be developed in any of the development languages that support Microsoft COM interfaces such as VB, VBA, Delphi, Visual J++, C, C#, C++ etc. This structure enables a licensee to develop his own components if/when required. Peresys does however prefer to be part of the development cycle or at least be involved in the quality assurance of newly developed components. If a client requires a new non-proprietary component which would enhance the product for the broader customer base we will deliver the component free of charge, subject to all the requirements being accurately documented and made available to Peresys.

Current components include:

- **Connectivity**

- ADO & ODBC
- MAPI and POP3 (E-Mail)
- IBM MQ Series and MSMQ
- FTP
- TELNET

- **Database**

- Database read and append
- Stored Procedure (execution)
- SQL read (SQL script execution)

- **Microsoft Office**

- MS Word write
- MS Excel read, write and macro
- MS Access read

- **File Manipulation and Transfer**

- Text file read and write
- XML read and write
- FTP read and write

- **Formatting & Parsing**

- 15 components comparable to VB string functions like mid, left, concatenate, trim etc.

- **Message Queuing**
 - IBM MQ Series (read and write)
 - Microsoft Message Queue (MSMQ) (read and write)

- **Connex Specific**
 - Decision Component (utilizes scenario type statements)
 - Procedure Command (ability to execute one procedure from within another)

- **Miscellaneous**
 - Command Shell (executes DOS commands)
 - Regular Expression
 - Zip Write
 - HTTP read and write
 - Telnet
 - Scripting (Creation of VBScript or JavaScript in the unlikely circumstance that the existing Connex components can not satisfy the requirements)

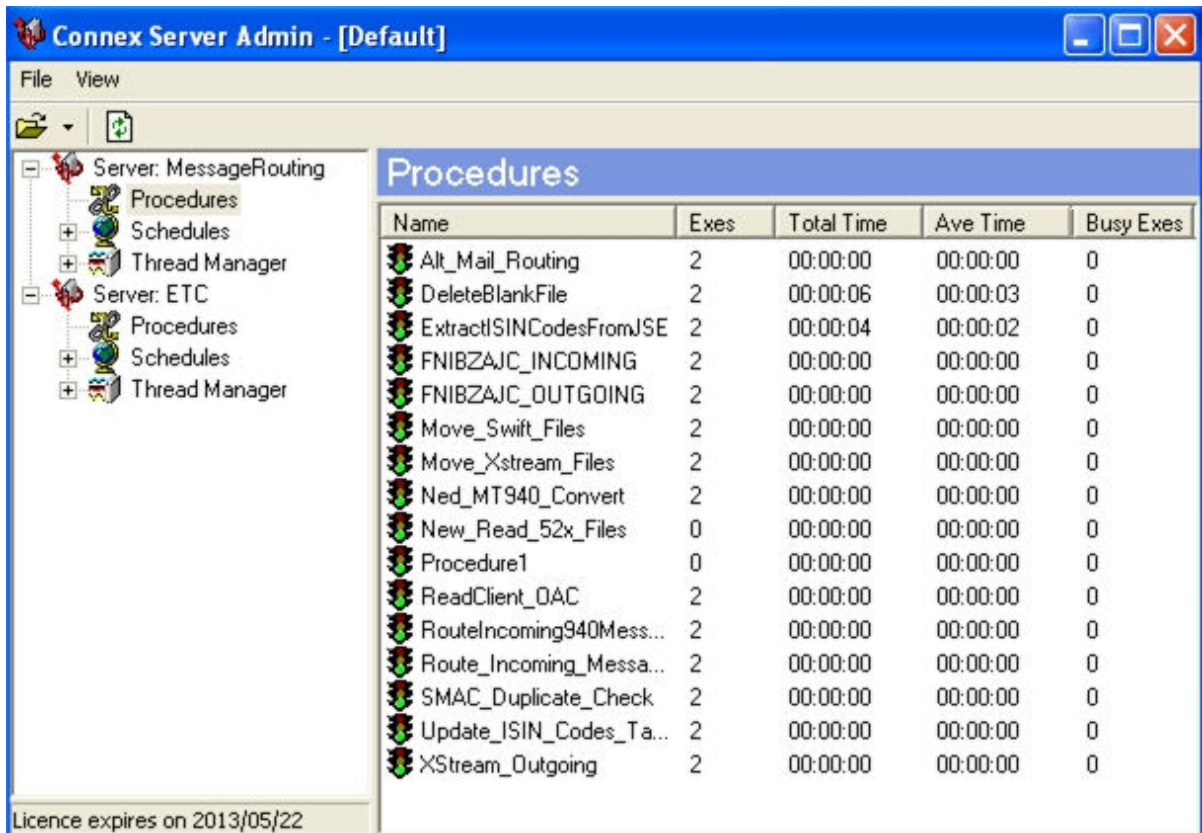
- **Financial**
 - SWIFT FIN
 - OASYS Global
 - HiPortfolio (Asset Management Administration)
 - SMAC (Asset Management Administration)
 - BDA (JSE Securities Exchange system)
 - FMC Net (Electronic Trade Data Communication)
 - FMC Pacer (Portfolio Management & Accounting)
 - CAMS (Banking system)
 - CATS (Banking system)
 - Nedinform (Banking system)
 - CashFocus (Banking system)
 - ABN AMRO Mellon (Custody System)
 - Citibank (Custody System)

- **CRM** (Client Relationship Management)
 - Goldmine
 - Siebel

Enhancements and expansions to the existing component suite are typically driven by client requirements and conformance to any modifications to public messaging standards.

Server Admin

Connex Server Admin is the process runtime engine, which executes the scheduled processes contained in Connex solution files:



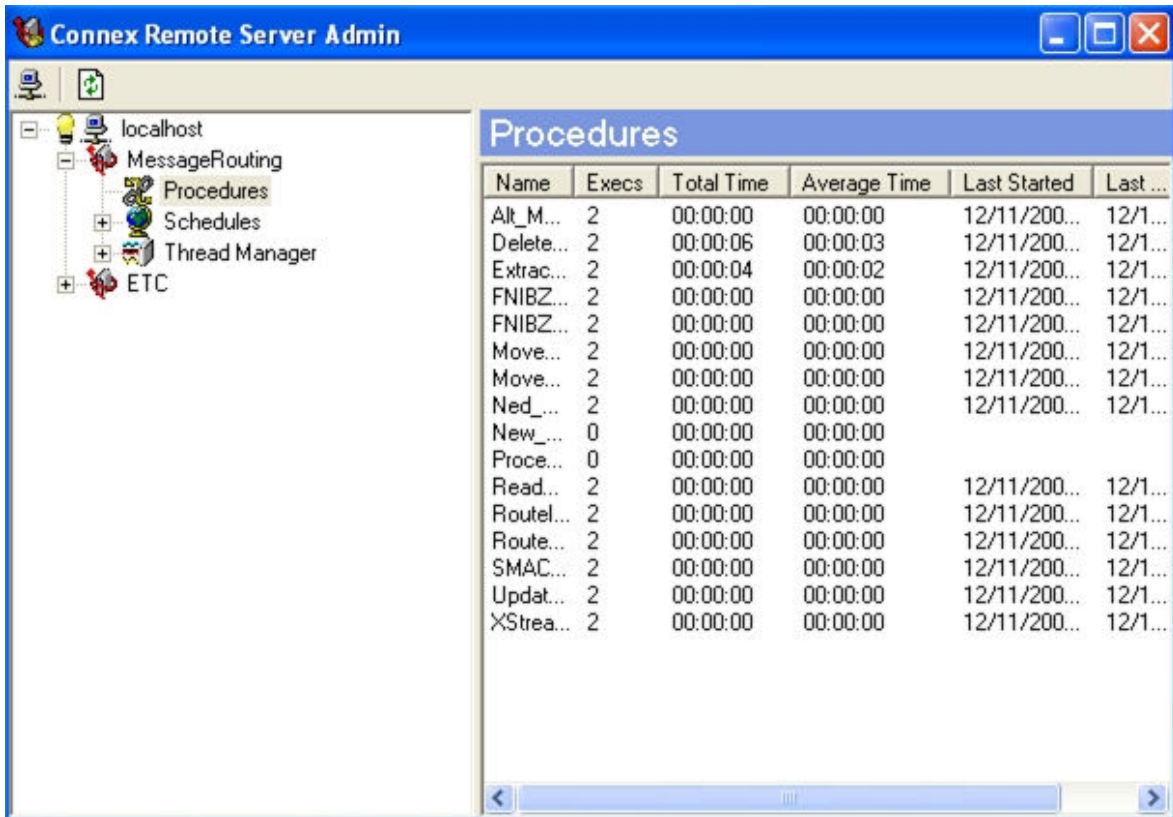
Server Admin has a visual interface where solution files can be added or removed at any time. It acts as an orchestration engine where procedures are concurrently executed according to the pre-configured schedules. These executions may be concurrent due to the multi-threaded architecture. Scheduling encompasses most options from a single execution on a specific date to continual executions every few seconds.

The user may also execute schedules or procedures manually or choose to deactivate a specific schedule or procedure within each solution. Post execution information includes a history of execution times, process duration and a detailed system log.

More than one instance of Server Admin may run on one machine with each instance managing numerous CNX files. This allows for a different configuration for each instance of Server Admin.

Remote Admin

Because physical access to the Server Admin component is often not possible, Connex Remote Admin provides a mechanism for remotely administering the various instances of Server Admin according to pre-configured user rights.

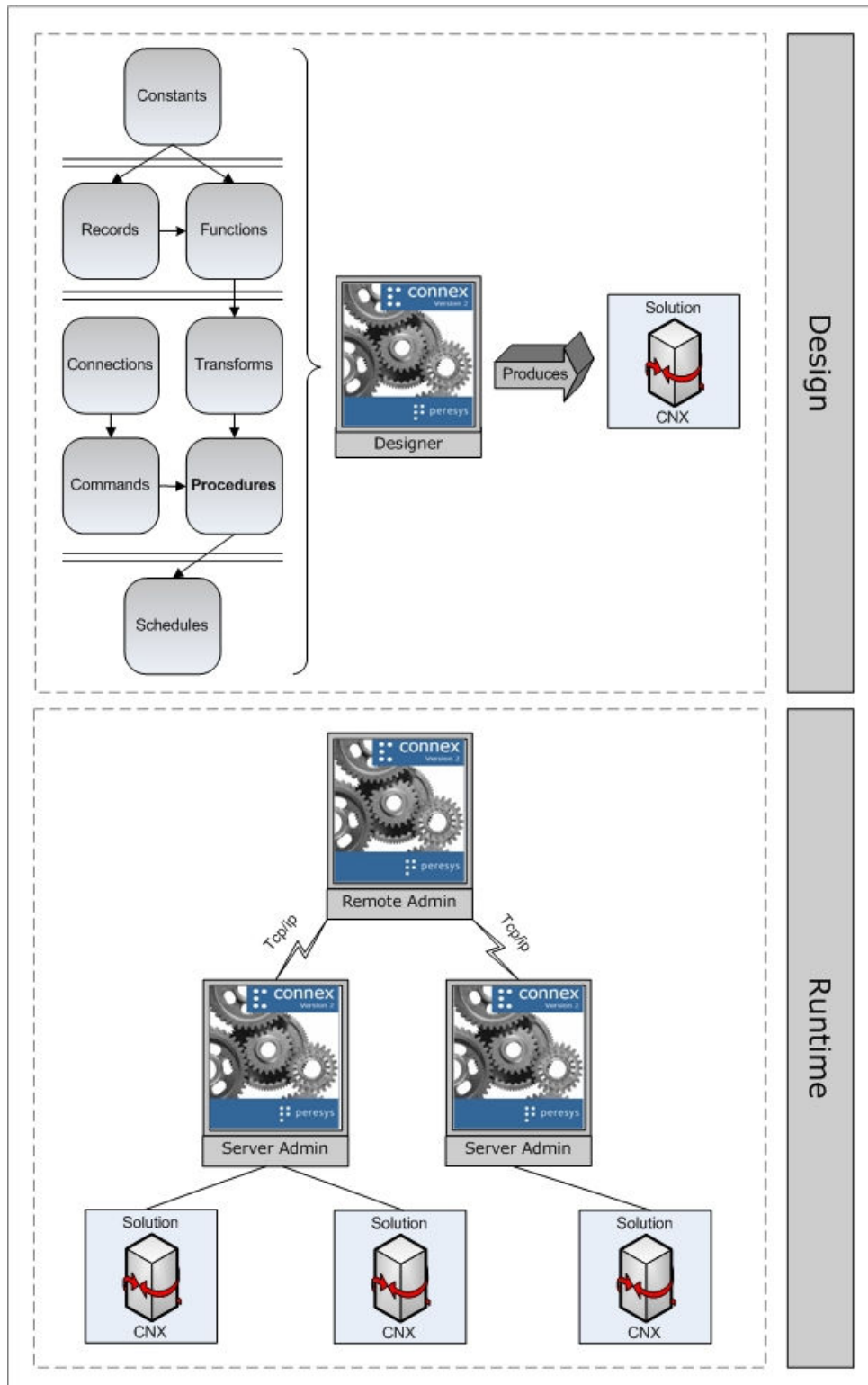


The screenshot shows the 'Connex Remote Server Admin' application window. On the left is a tree view under 'localhost' with nodes for 'MessageRouting', 'Procedures', 'Schedules', 'Thread Manager', and 'ETC'. The 'Procedures' node is selected, and the main area displays a table with the following data:

Name	Execs	Total Time	Average Time	Last Started	Last ...
Alt_M...	2	00:00:00	00:00:00	12/11/200...	12/1...
Delete...	2	00:00:06	00:00:03	12/11/200...	12/1...
Extrac...	2	00:00:04	00:00:02	12/11/200...	12/1...
FNIBZ...	2	00:00:00	00:00:00	12/11/200...	12/1...
FNIBZ...	2	00:00:00	00:00:00	12/11/200...	12/1...
Move...	2	00:00:00	00:00:00	12/11/200...	12/1...
Move...	2	00:00:00	00:00:00	12/11/200...	12/1...
Ned_...	2	00:00:00	00:00:00	12/11/200...	12/1...
New_...	0	00:00:00	00:00:00		
Proce...	0	00:00:00	00:00:00		
Read...	2	00:00:00	00:00:00	12/11/200...	12/1...
Routel...	2	00:00:00	00:00:00	12/11/200...	12/1...
Route...	2	00:00:00	00:00:00	12/11/200...	12/1...
SMAC...	2	00:00:00	00:00:00	12/11/200...	12/1...
Updat...	2	00:00:00	00:00:00	12/11/200...	12/1...
XStrea...	2	00:00:00	00:00:00	12/11/200...	12/1...

It minimizes risk by restricting access to Server Admin and assigning the appropriate rights to each connecting user. Rights exist around loading and removing solution files, starting, stopping, activating or deactivating schedules as well as executing procedures. Furthermore certain users may only be allowed to connect to a specific instance of Server Admin, thus adding another level of security.

Module Interaction



Connex Implementation

A typical Connex integration solution can be formulated as follows:

- A need for **application integration** or **business process automation** is identified. These processes are typically resource intensive, time consuming and error prone due to manual intervention.
- **Perform** a full business / systems analysis of the environment and processes to be automated and integrated. This includes identifying all related systems together with a detailed analysis of the required data flows between these systems. It is essentially a process of identifying data that will be injected into and extracted from the middleware stream. This stream of data will be distributed from source to destination in accordance with a set of business rules defined within Connex. The analysis should include all checks and balances to be placed on the movement of data between systems.
- **Determine** if all software connectors (components), that are required to exchange data with the middleware stream, are available. In-house systems are potential areas where software connectors may not exist. Any system that exposes some form of data import and export functionality is a candidate for Connex to interact with. Subject to the standard criteria including reusability, a COM component could be created and added to the Connex suite of components.
- **Model** the process within the Connex design environment. The initial modelling will consist of setting up input and output data definitions required for the identified systems. This will then be expanded by defining transformation processes that will prepare data for transmission to and from the systems. Once this is complete, data processing procedures are defined. These will form the basis of interaction between the systems using the defined transformation processes.
- Procedures can be automated by the configuration of **schedules**, which in turn manage the executions at user definable times and / or intervals. Before setting up schedules, the overall process flow of the solution is considered. For example, one procedure might be dependant on the successful completion of another. Also, certain business processes might run at set times, which could have an impact on the availability of required data.
- Once tested and implemented in a production environment, Connex will seamlessly execute in the background given that all incoming data is provided timeously and in the correct format. Authorized users can interact with the **orchestration engine** (Server Admin) via the Remote Admin facility. Depending on the configuration, users are allowed to add or remove solution (.cnx) files, view the log file, manually execute procedures and de-activate schedules or procedures. End users interact with the solution via the Message Viewer application which gives them controlled database access to the information flowing in and out of the system.

Future Releases

Connex is evolving and we continue to grow and improve the product on the back of requirements submitted by our customers. Some of the features to look forward to include:

- **Event driven execution** – Ability to process data based on events. For example reading from a message queue as soon as a message arrives or processing a file when it arrives in a pre-defined directory.
- **Solution file stored as XML** – There are several benefits to this approach including facilitating change management and source control.
- **Merging tool** – This tool will provide functionality for porting objects (and all dependant items) from one solution file to another. This will greatly enhance re-usability and also facilitate the interaction between development, test and production environments.
- **New logging mechanism** – Ability to configure the logging requirements. It can be done at a very high level where minimum information is logged per solution file or alternatively at a detail level where information is logged per component (residing within a procedure within a solution file).
- **New Swift component** – Over and above creation and parsing ability, new functionality includes the validation of each message according to Swift standards and the conversion of Swift to XML or XML to Swift. The use of XML structures enables us to cater effectively for complex message structures/hierarchies like repeating blocks within blocks.
- **New XML component** – The XML Component was redesigned to more effectively handle repeating and recursive elements. This redesign is a powerful foundation for components like Swift or any others that make use of complex data structures. It supports different modes of execution, standard which is an easy and simple mode for standard and simple XML files, and recursive mode for more sophisticated XML files, which can orchestrate on a per-node basis to sub command branches. The new XML component creates a foundation for any solution which requires recursion or repeating data, without compromising the simplicity and ease of use of the standard Connex framework. The component also has validation by means of DOM support.