www.skills-1st.co.uk

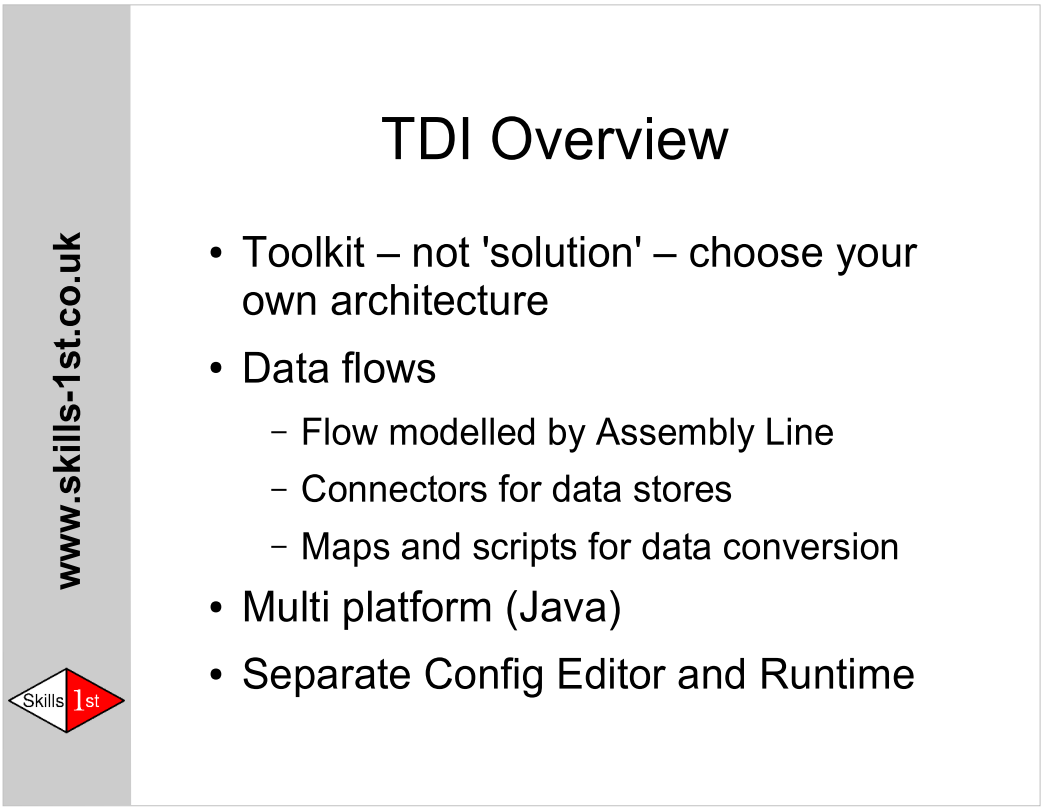
TDI v7

Tivoli Directory Integrator v7

Workshop

Andrew Findlay
Version 1.1
January 2011

A metadirectory system provides for the flow of data between one or more directory services and databases, in order to maintain synchronization of that data (Wikipedia).
TDI is one of the most flexible metadirectory products on the market. It does not impose a fixed structure on the solution, and is not limited to handling directory-like data.



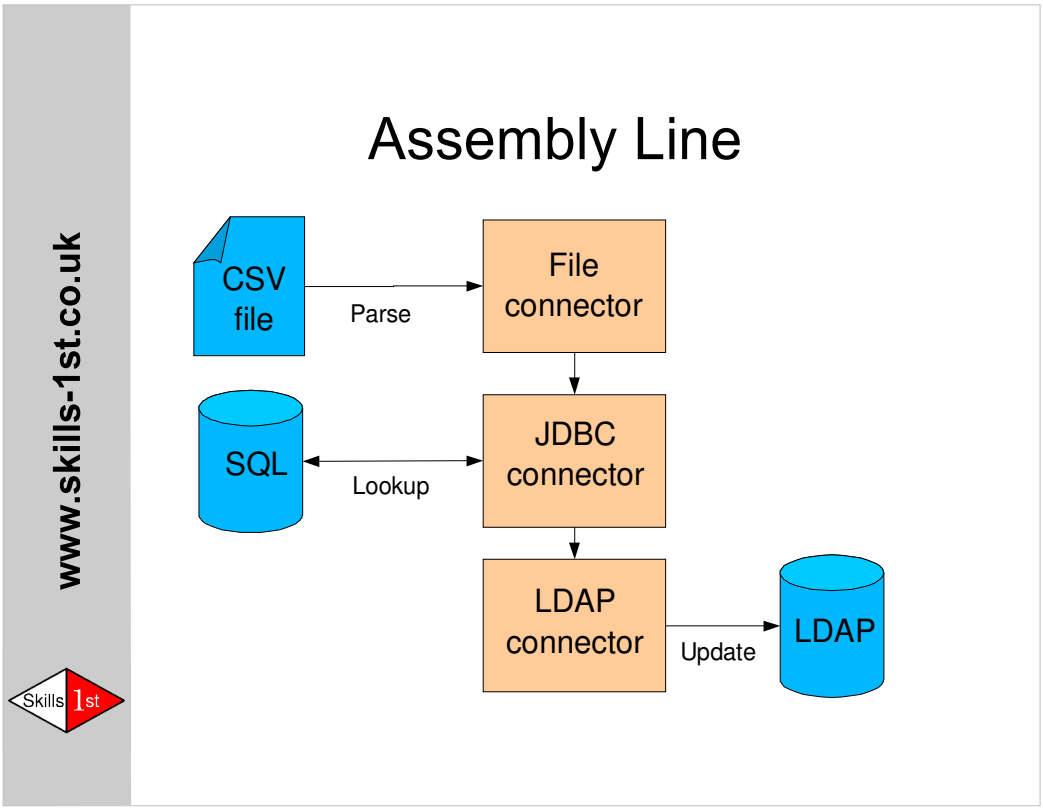
TDI is like a big toolbox: it has components for connecting to all sorts of things, for manipulating data, and for producing reports. It also has features to help you get started on a job quickly – schema discovery, data browsers etc.
TDI will run on almost any current server operating system.
The graphical Config Editor is distinct from the runtime system, so you can build applications to run on machines with no graphics.

www.skills-1st.co.uk

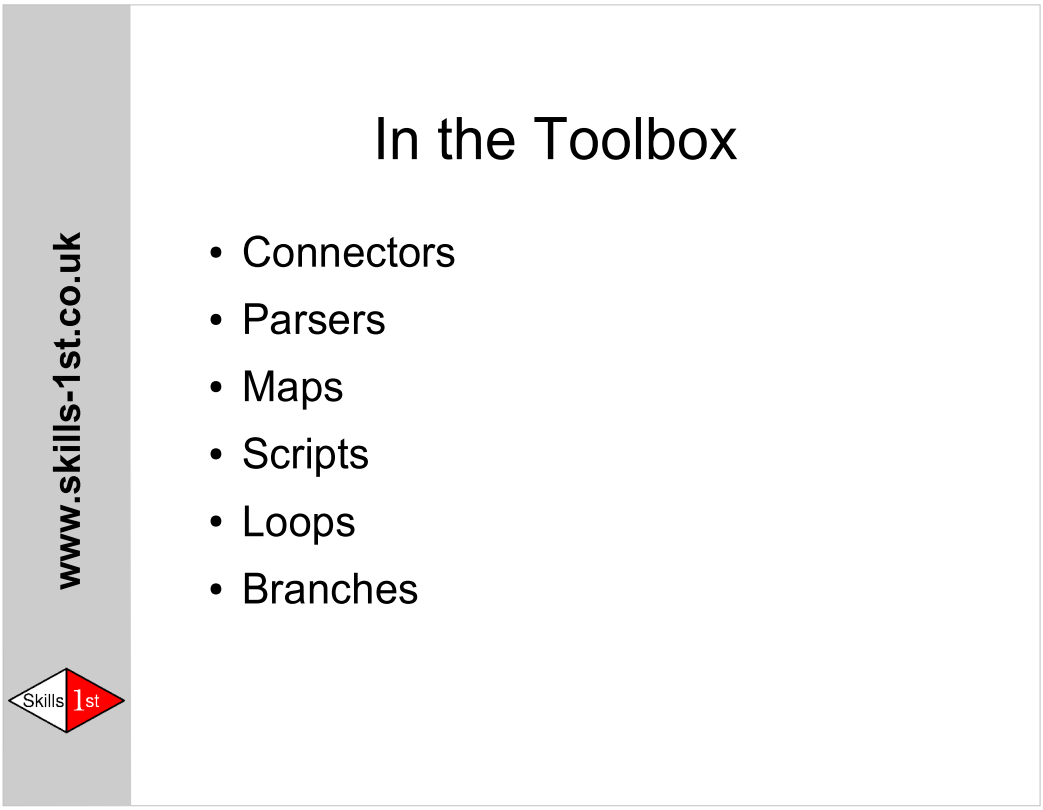
TDI Data Model

- Process one entry at a time
- No fixed relationship between entries
- Entries are not named
- The Work Object – like an LDAP entry
  - Named attributes
  - Zero or more values per attribute
  - Values can be of any type

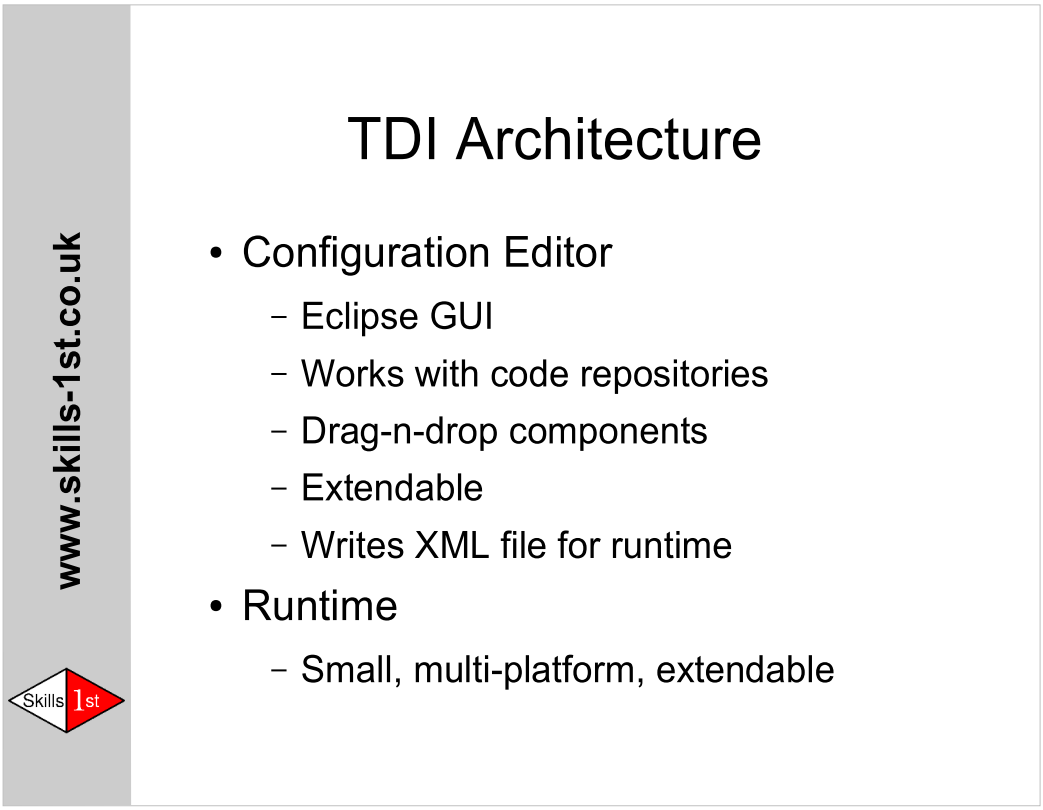
An Entry is a collection of data describing some real-world object such as a person, an account, or a message.
If the name of an entry is important, it can be carried in an attribute. TDI itself does not have a naming scheme for entries as it does not store any.



Assembly Lines model data flows. Here we read lines from a CSV file, parse them so that columns become attribute values, then pass the resulting entry down the line. The second connector adds value to the entry by looking up data in an SQL database. The third connector writes the whole entry into an LDAP directory.



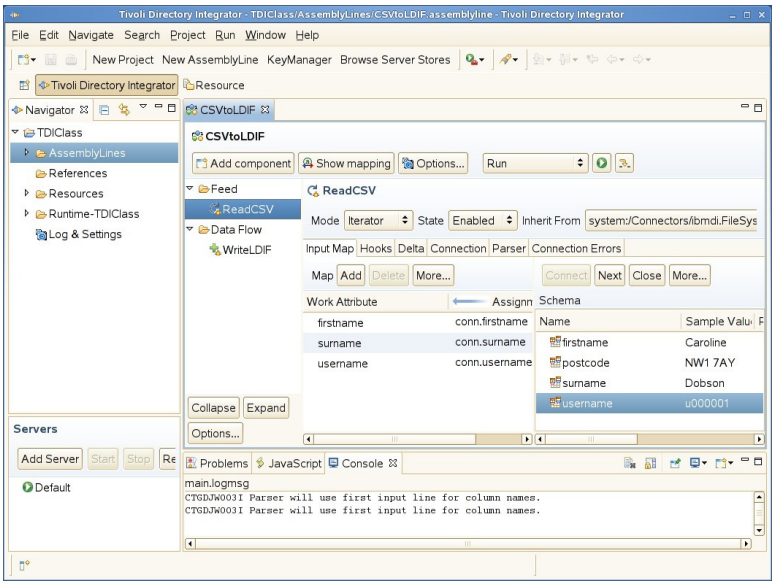
Connectors link to various data sources and destinations. They hide the details of making the connection and present a unified interface to the assembly line.
Parsers convert to and from various (usually text-based) representations of data – CSV, XML etc.
Maps change the names and/or values of attributes.
Scripts can do anything ☺
Loops and branches provide flow control within the assembly line.



The Config Editor (CE) has been based on Eclipse since TDI 7.0. This allows standard software development tools to be used with TDI configurations.

www.skills-1st.co.uk

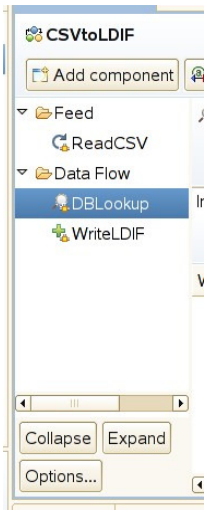
The Config Editor



You need a large screen for this!

www.skills-1st.co.uk

Assembly Line View

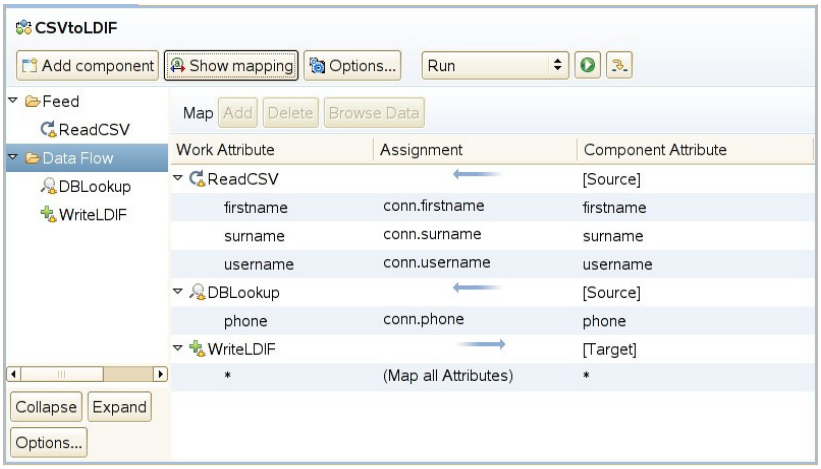


- Can expand items
- Drag things up and down
- Add and delete

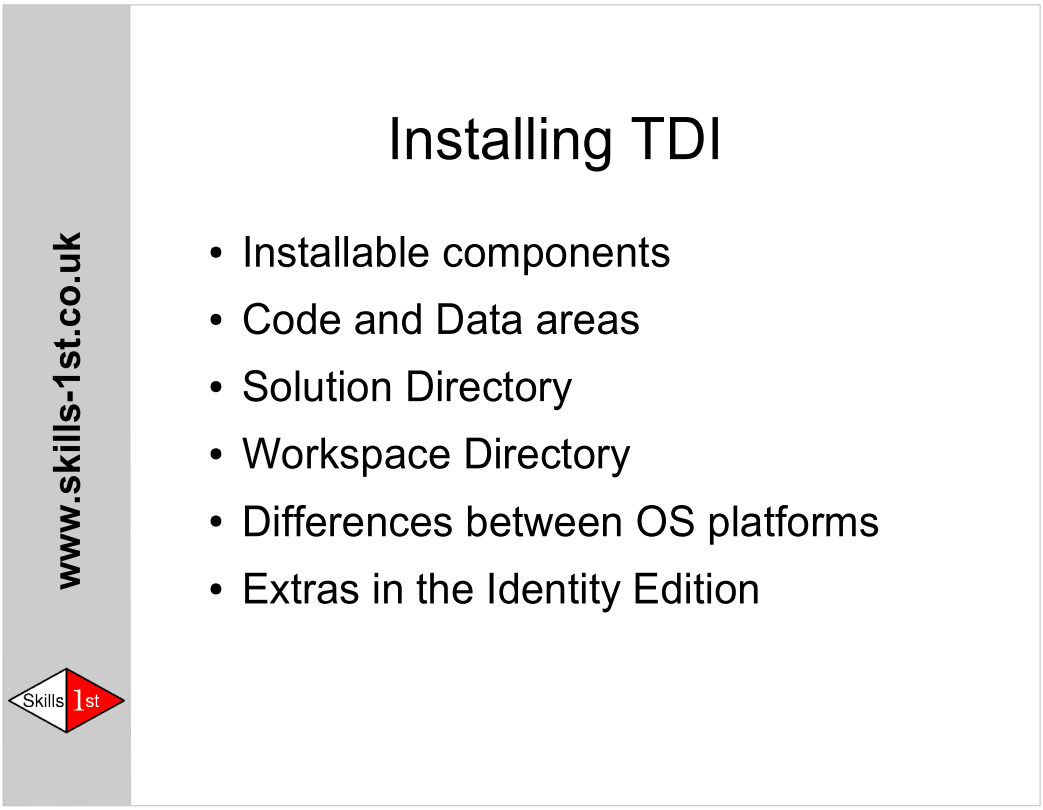
Entries flow down the assembly line from top to bottom.

www.skills-1st.co.uk

Show Mapping



This view gives a quick overview of where each attribute comes from and where it is used.
Be aware that things done by scripts will not show up here.



Components
- Runtime Server
- Config Editor
- Eclipse Update Site for CE
- Java API docs
- Examples
- Local Helpfile hosting (need to download the files separately)
- Integrated Solutions Console (web server)
- Admin and Monitoring Console (web app)
- Java Runtime Environment

Solution Directory holds runtime config
Workspace Directory holds TDI source files for Eclipse

Some OS platforms do not support the config editor (mostly mainframes)

Identity Edition includes password capture modules and has a different pricing structure.

www.skills-1st.co.uk

The workshop environment

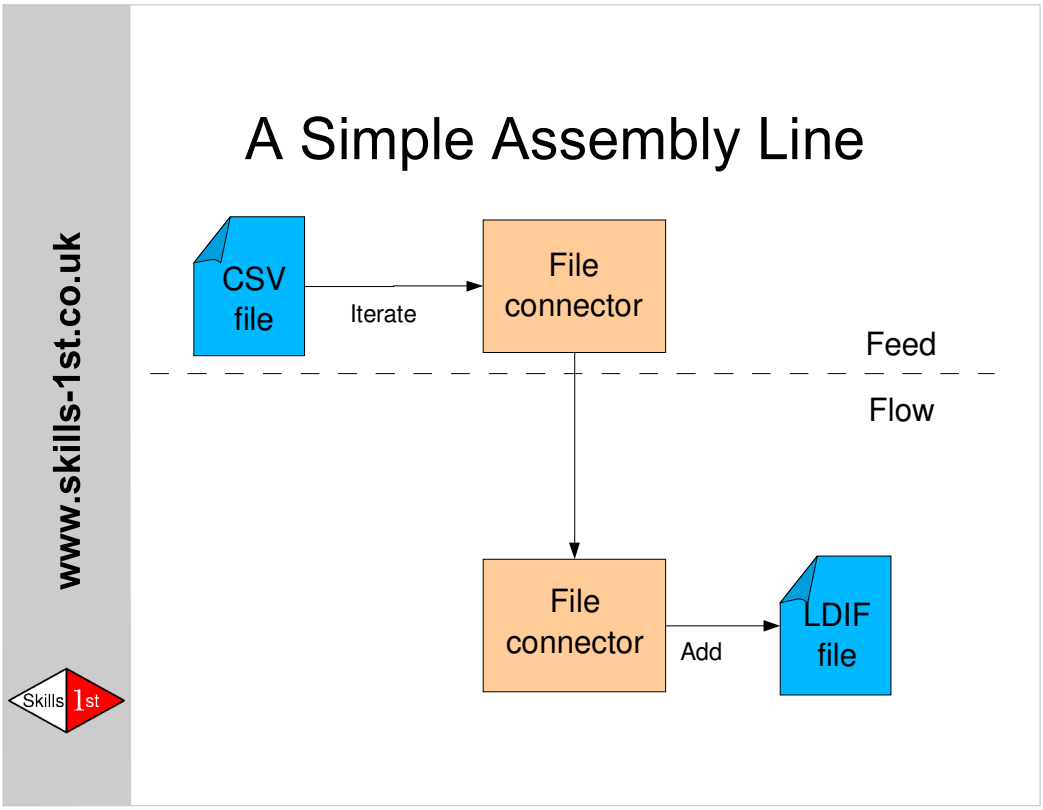
- VMWare
- SuSE Linux: SLES 10 eval
- *student* account
- TDI v7.1 Identity Edition eval
- Datafiles
- MySQL database
- LDAP servers

www.skills-1st.co.uk

Skills 1st

Exercises 1,2,3

- Login and explore
- Install TDI
- Start the Config Editor

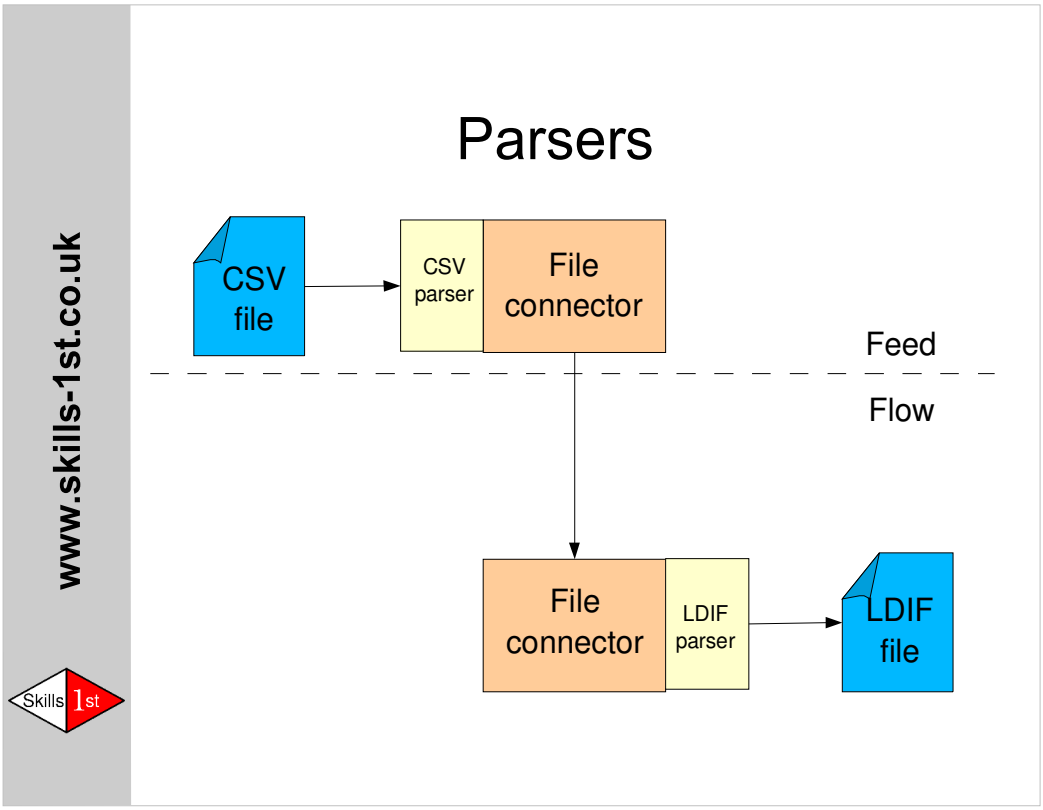


We want to convert from this:

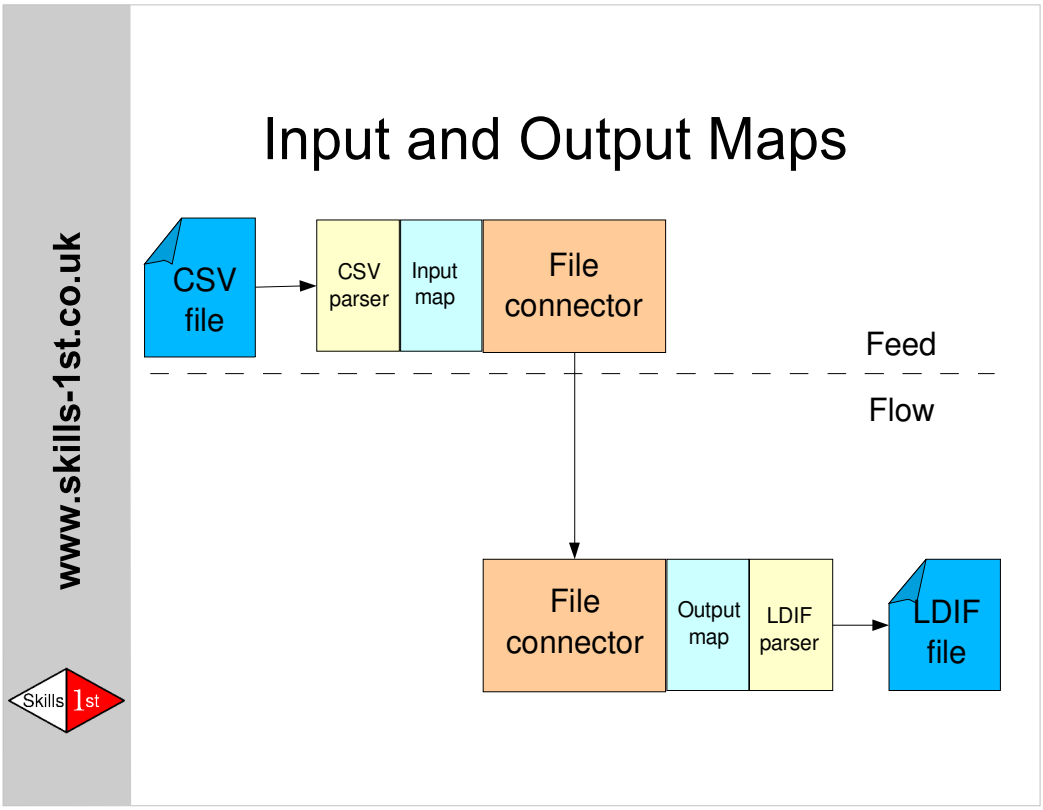
```
username,firstname,surname,postcode
u000001,Caroline,Dobson,NW1 7AY
u000002,Clive,Rhodes,
```

To this:

```
dn: uid=u000001,dc=people,dc=example,dc=org
postalcode: NW1 7AY
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
uid: u000001
sn: Dobson
telephoneNumber: +44 987 000001
cn: Caroline Dobson
```



Parsers in TDI can generate text as well as parse it.



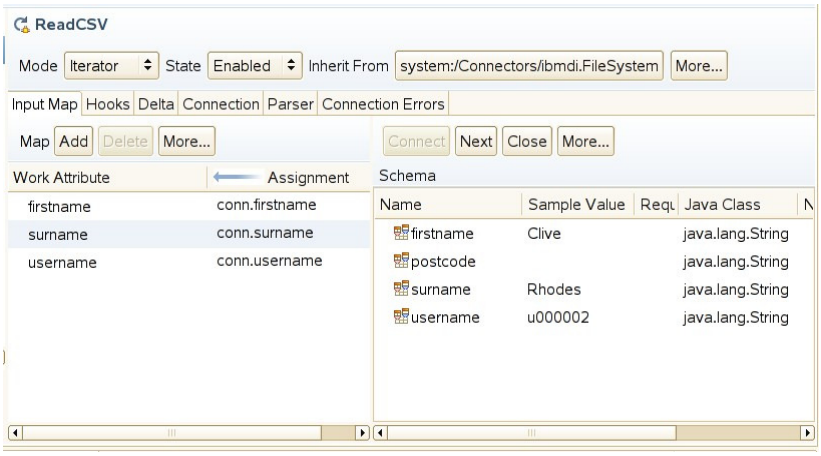
Maps can:

- Change the names of attributes
- Create new attributes
- Change the values of attributes

e.g. in this case we need to map the 'username' column to the 'uid' attribute, and use both 'firstname' and 'surname' data to build the 'cn' attribute.

www.skills-1st.co.uk

Input Map and Schema Discovery



Schema discovery is a very powerful feature available once the basic connection has been configured. The CE connects to the data source and extracts sample data to help you set up the maps. Drag schema items into the work attribute to set up simple mappings.

www.skills-1st.co.uk

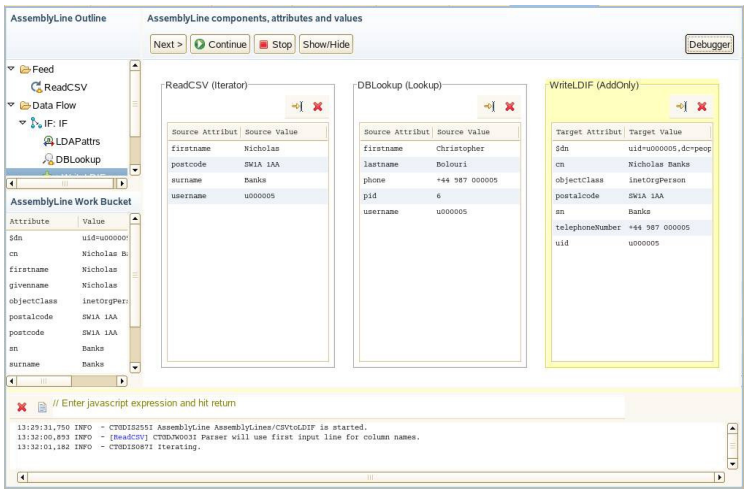
Skills 1st

Exercise 4

- Create an Assembly Line
- Read CSV file
- Write LDIF file

www.skills-1st.co.uk

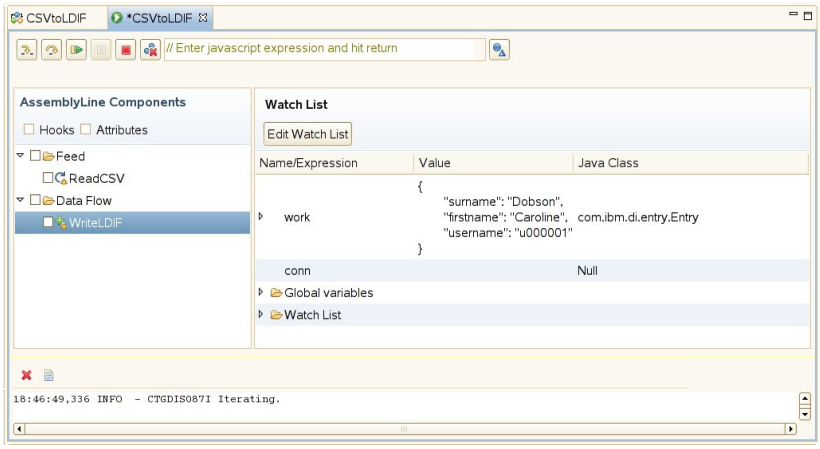
Data Stepper



New with TDI 7.1
This is now the default debugger view.

www.skills-1st.co.uk

TDI Debugger



- Single-step (over or into)
- Set breakpoints
- Examine data
- Evaluate JavaScript extensions

Change level of detail:

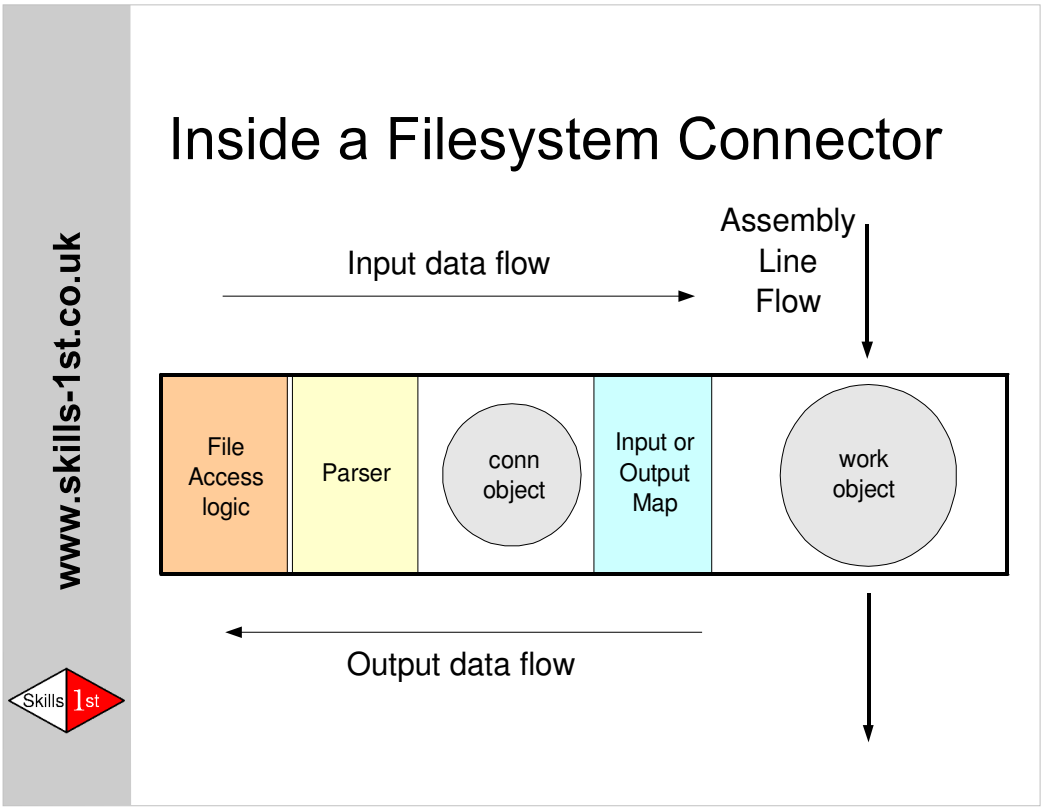
- Connectors / major AL components
- Hooks
- Input and Output maps
- JavaScript line-by-line

www.skills-1st.co.uk

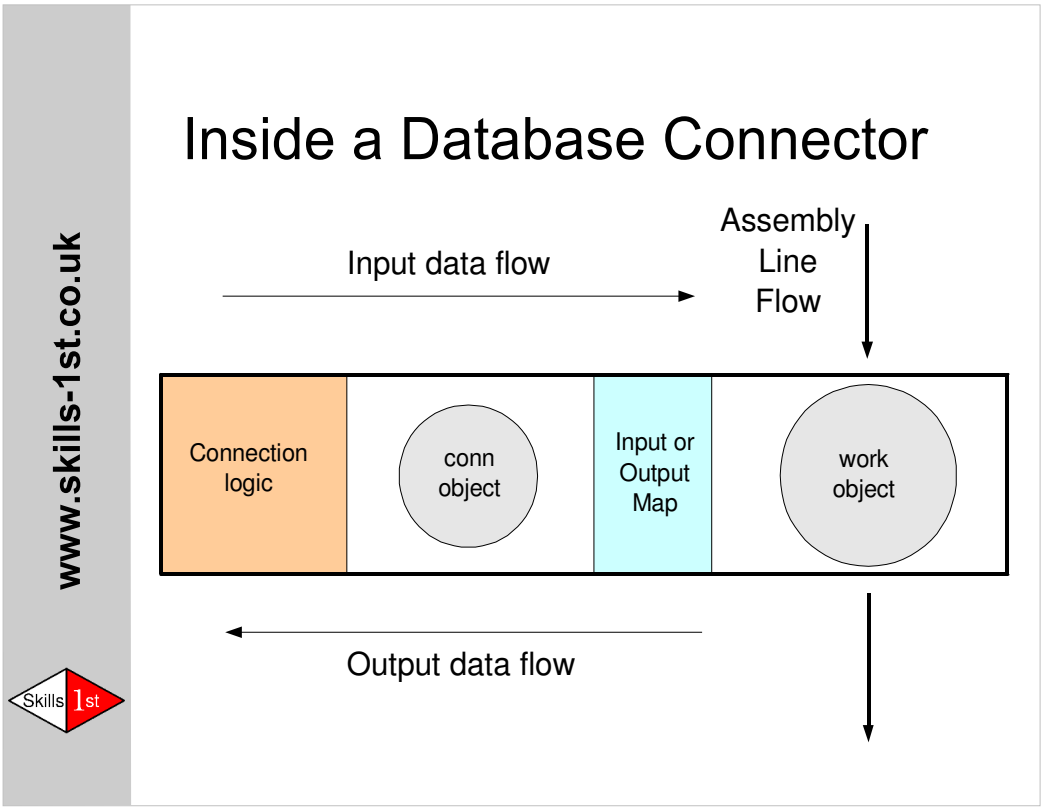
Skills 1st

Exercise 5

- TDI Debugger



This is why input maps reference items like **conn.surname** and output maps reference things like **work.surname**



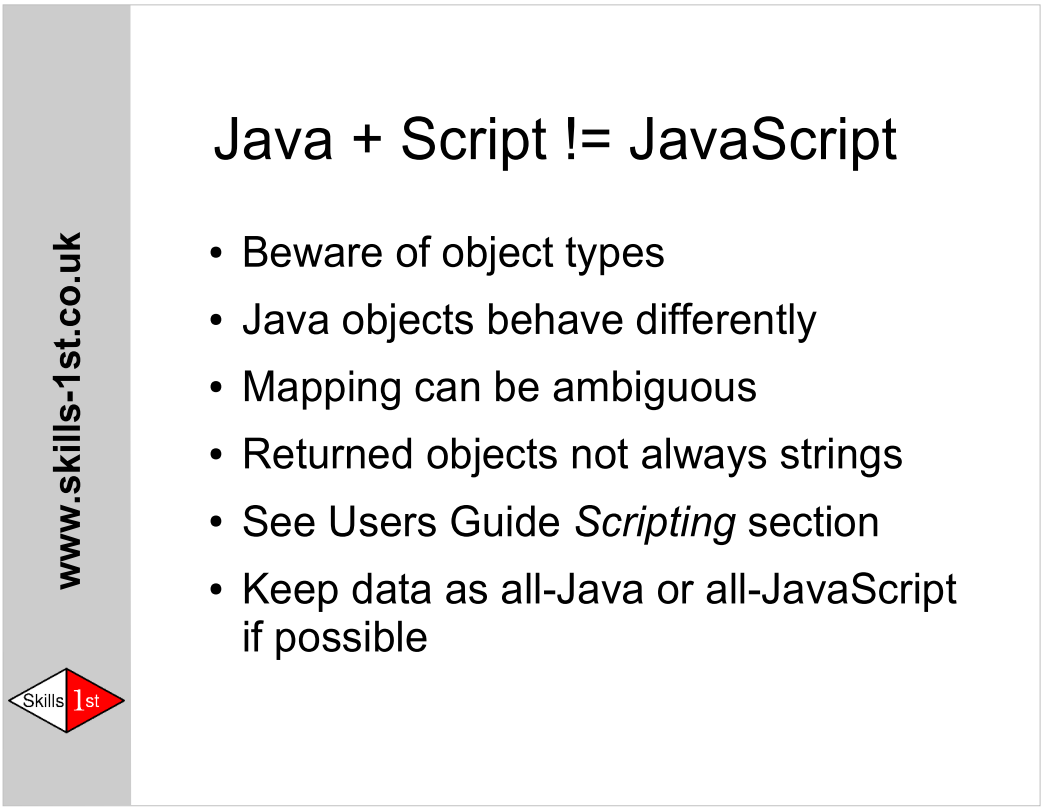
Database connectors do not usually have parsers, as the data sources present attributes and values in a structured form.

www.skills-1st.co.uk

Scripting in Maps

- Scripts can build new values
  `work.firstname + " " + work.surname`
- Scripts can be:
  - Expressions (as above)
  - JavaScript code
  - Text with substitution

The Eclipse code editor is context-sensitive. It does things like auto-adding closing quotes to text strings, closing parentheses when you type opening ones etc.



String comparisons can be particularly tricky. This form is safe as it is entirely in Java:

```
work.firstname.getValue().equalsIgnoreCase("Andrew");
```

This form is not safe:

work.getString(“firstname”) == work.getString(“targetname”);

(JavaScript ends up comparing the locations of two java.lang.String objects rather than the actual strings.)

Assembly-line:
Options -> Assembly Line Settings -> Null Behaviour
Connector:
Input Map -> More -> Null Behaviour
Attribute:
Right-click on attribute in map, select Null Behaviour

Definition of Null:
- Default
- Absent attribute
- Empty attribute
- Empty string or null value
- Specified value

Actions:
- Default
- Delete attribute
- Use null value
- Use empty string
- Use specified string
- Throw exception

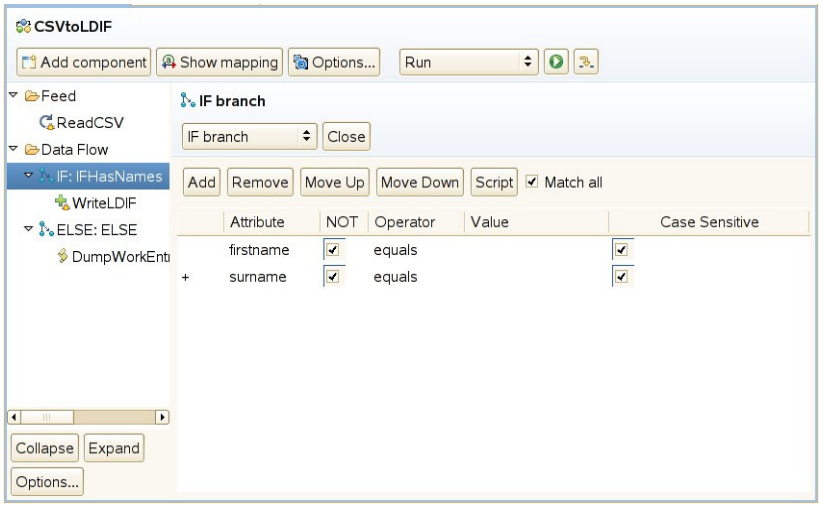
www.skills-1st.co.uk

Exercise 6

- Attribute Maps
- Improve your LDIF output

www.skills-1st.co.uk

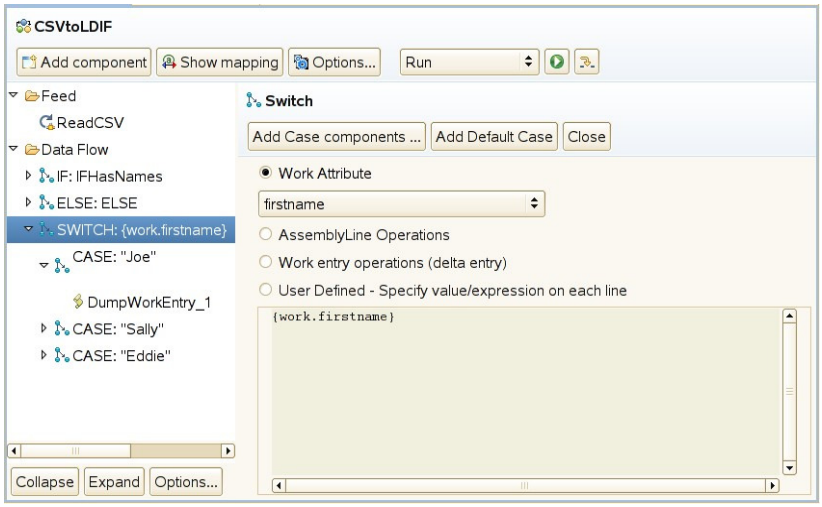
Flow Control: IF-THEN-ELSE



IF and ELSE are inserted as separate components
You need at least one component in the branch before you can drag anything else into it.
ELSE-IF is also available
Conditions built from AND or OR combination of rules.
Script rules also possible

www.skills-1st.co.uk

Flow Control: Switch/Case



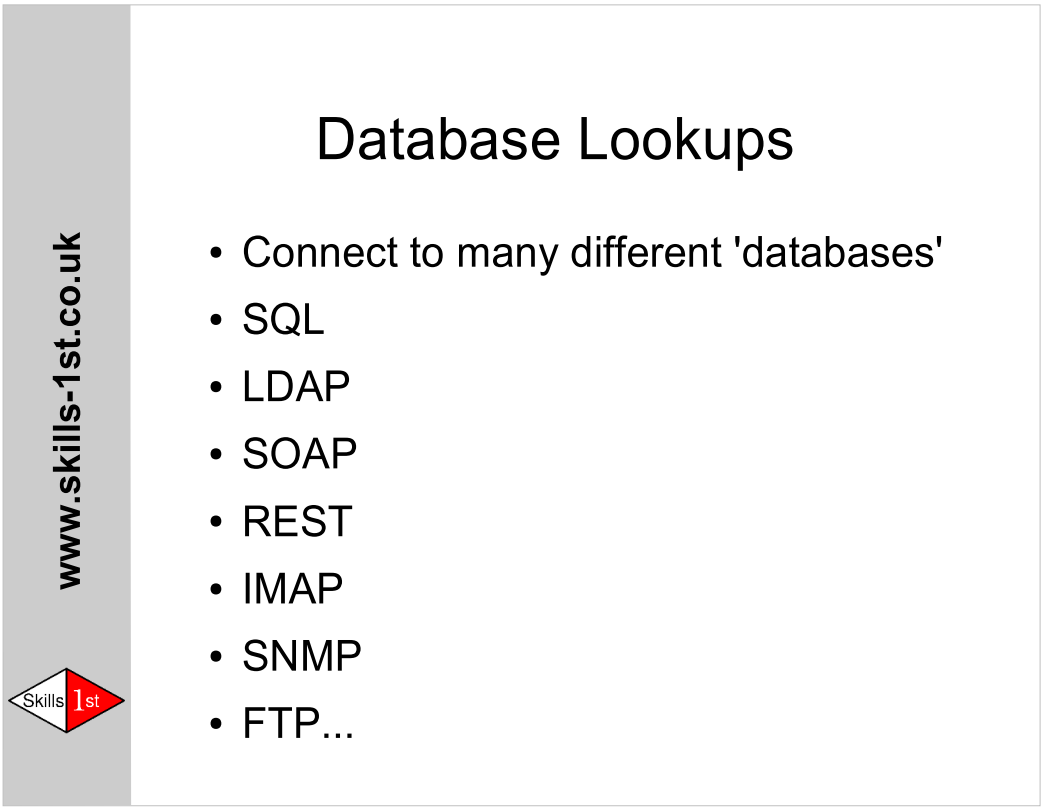
Add the SWITCH component and then use the *Add Case Components* button to add each case in turn.

www.skills-1st.co.uk

Skills 1st

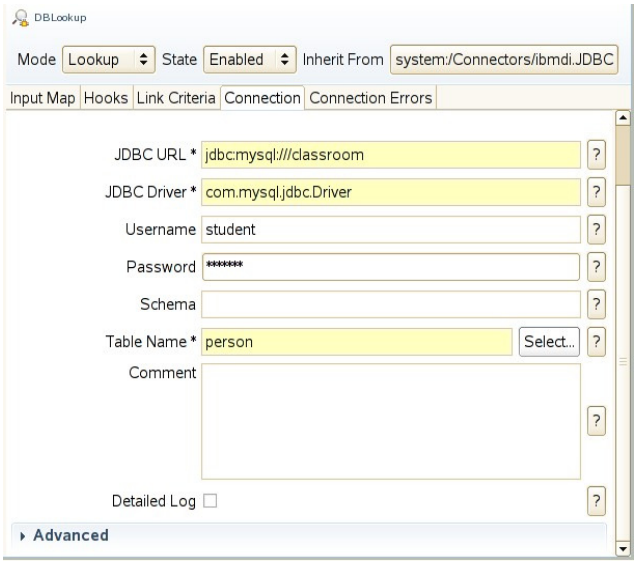
Exercise 7

- Use IF and ELSE
- Report entries with missing names



Many connectors can be used in Lookup mode: any data source where you can send a request and get a response based on it can be used this way.

JDBC Connector

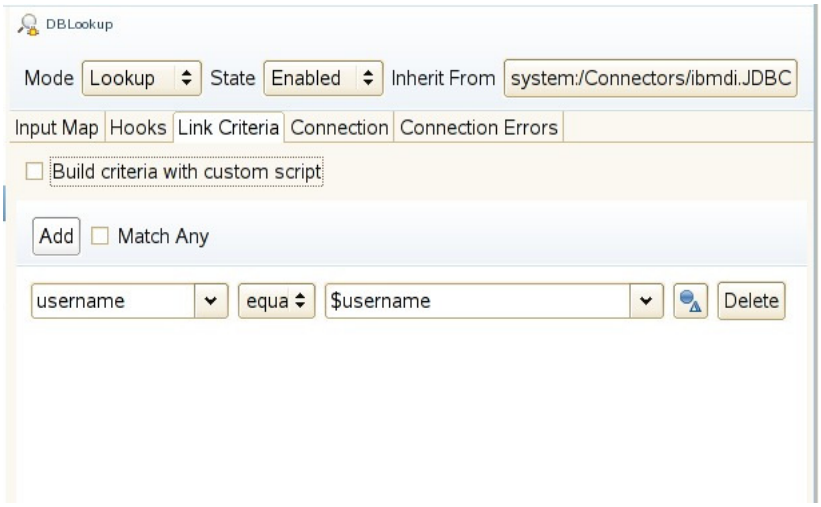


JDBC is the Java Database Connection framework. It can be used with many different data sources, though the most common ones are relational databases.
The database is identified by URL, and you need to specify the name of the driver library. You can add Java drivers if TDI does not already have the one you want.
A good test is to fill in the first few fields and then click the *Select* button beside the *Table Name* field. If the connection works you will get a drop-down list of tables to select from.

www.skills-1st.co.uk

Skills 1st

Link Criteria



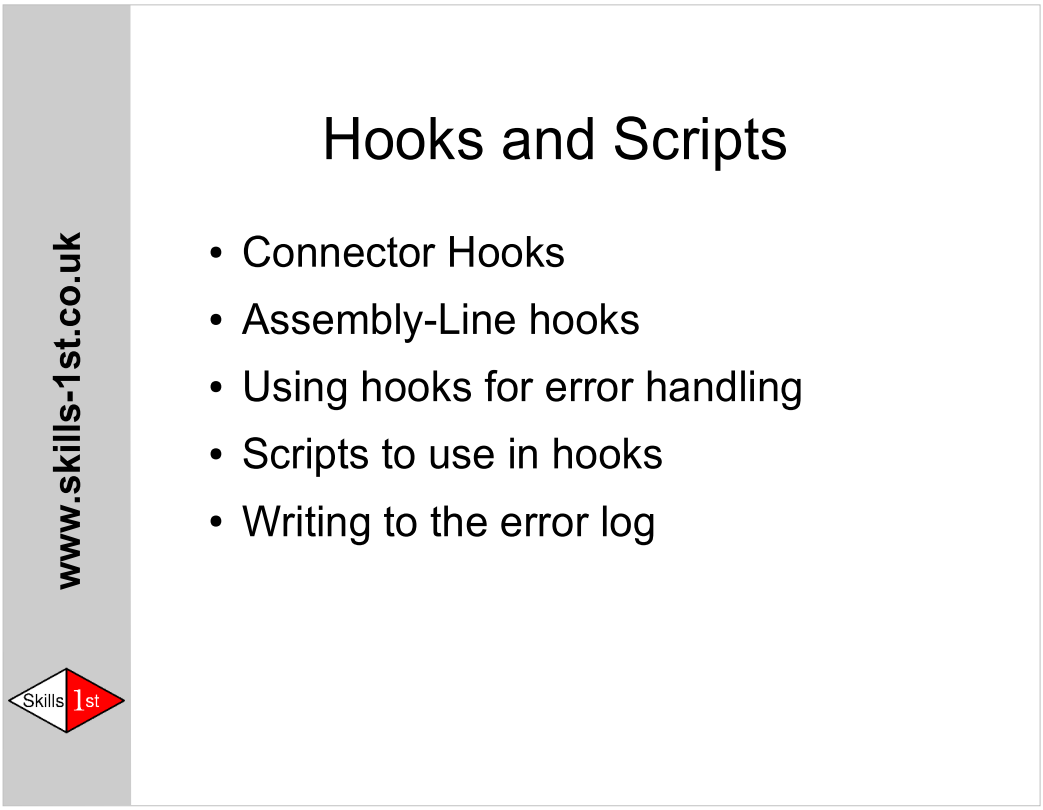
TDI uses Link Criteria to build database queries, so in many cases you will not even have to write a line of SQL. Exactly the same interface is used for LDAP and many other data sources.

www.skills-1st.co.uk

Skills 1st

Exercise 8

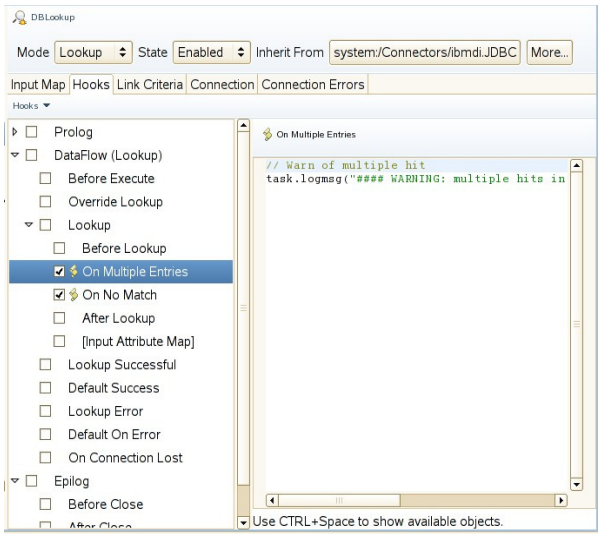
- Configure JDBC for MySQL
- Lookup phone numbers in SQL database



Hooks are places where you can hang scripts to do things that TDI does not have built-in functions for.
In a real-life TDI implementation you will probably spend more time working on hooks and scripts than on anything else.

www.skills-1st.co.uk

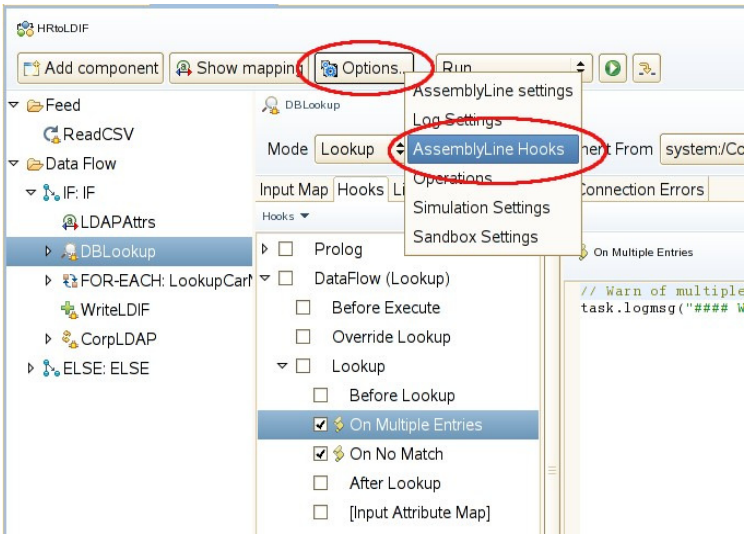
Connector Hooks



Some hooks are called when the connector is initialised, others are called for every work object processed. Many are only called when specific conditions occur: these are mostly used for exception handling.
The hooks are called roughly in the order shown in the display, but you should consult the detailed flowcharts and individual connector descriptions if the sequence is critical. In general it is best not to depend on the sequence, particularly in the setup and teardown phases.

www.skills-1st.co.uk

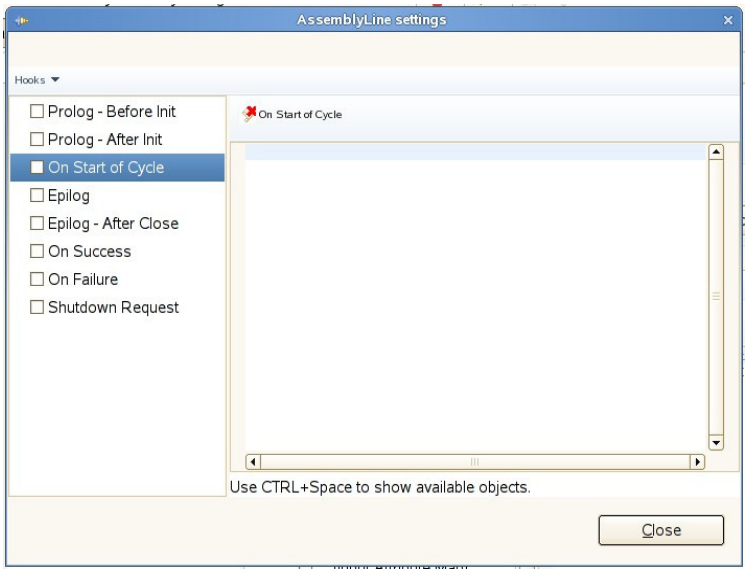
Assembly Line Hooks



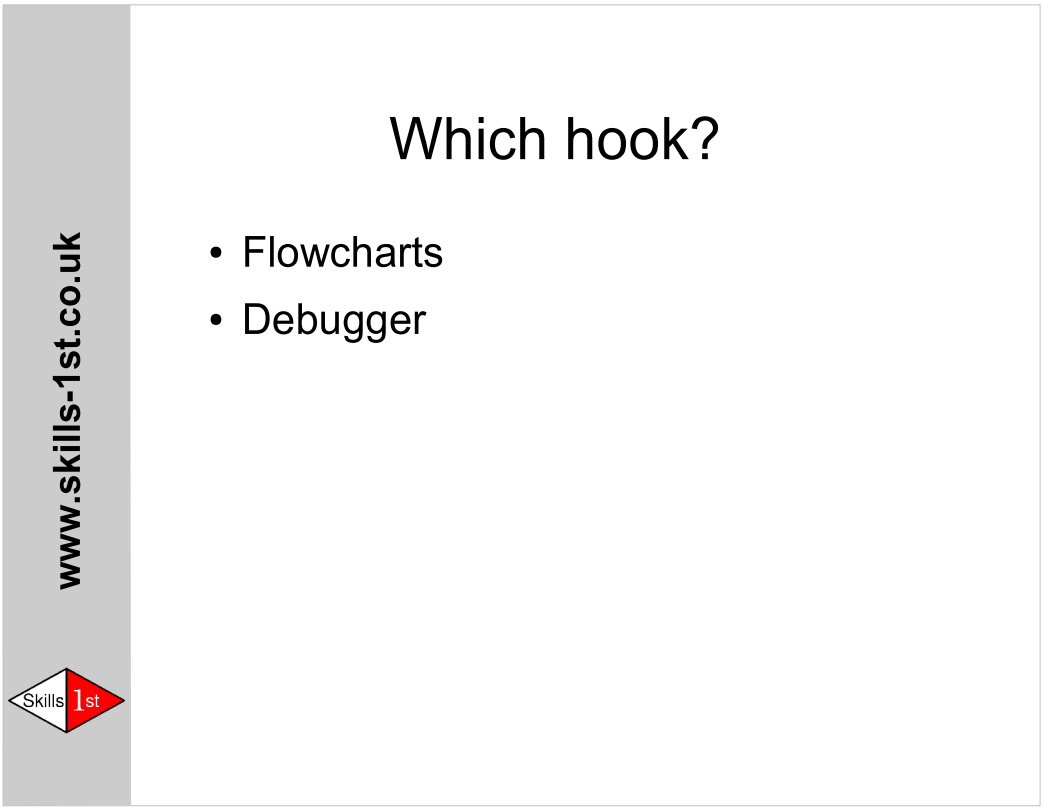
This shows how to access the AL hooks in TDI 7.x

www.skills-1st.co.uk

Assembly Line Hooks



Assembly-line hooks are mostly concerned with initialisation and tear-down.



The Assembly Line Flow Diagrams used to be an appendix to the Reference Guide but are now a separate PDF document.

As the debugger now exposes hooks, a very good way to understand the flow of execution is simply to step through your assembly line with all the hooks visible.

Available objects

- Ctrl-space in script editor
- Typically:
  - work
  - conn
  - error
  - system
  - task
  - All named components in AL

The list of available objects varies depending on where the script is used.
Mapping scripts usually operate between the *work* and *conn* objects.
Most of the useful objects come from the Java level. TDI-specific objects and methods are described in the *JavaDocs* – web pages that are auto-generated from the TDI source code. The JavaDocs can be found in the installation area, typically /opt/IBM/TDI/V7.1/docs

www.skills-1st.co.uk

Skills 1st

Exercise 9

- Hooks and Scripts
- Basic error handling
- Missing Data
- Multiple hits

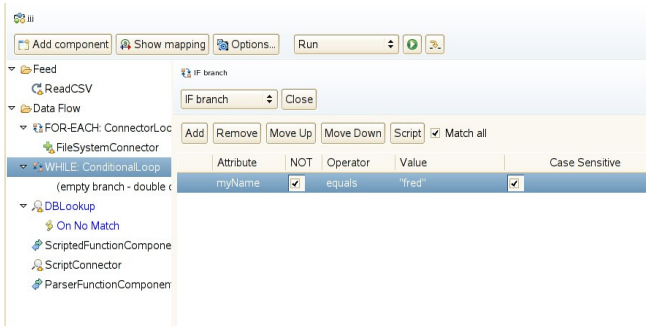
Loops

- Conditional loops
  - While <condition> do...
- Attribute-value loops
  - Handle each value separately
- Connector loops
  - Handle each entry of a multi-hit result

www.skills-1st.co.uk

Conditional loops

- Loop continues while condition is true
- Condition set just like IF-THEN

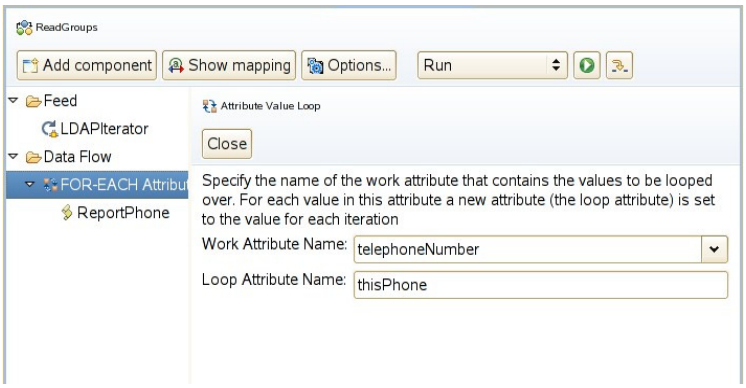


This is the classic loop found in most programming languages. Early versions of TDI (before 6.0) did not have loops, so solutions developed for those versions had to use scripts when looping was required.

www.skills-1st.co.uk

Attribute-Value loops

- Iterate through values of one attribute
  - e.g. Members of an LDAP group

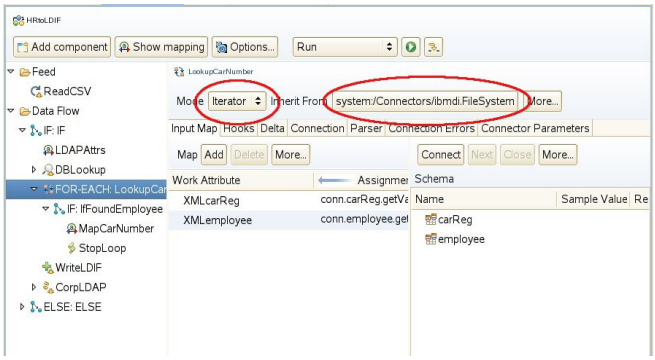


This gives you a way to process each value of a multi-valued attribute separately. The whole attribute remains available as the current value is held in a new attribute while the loop is in progress.

www.skills-1st.co.uk

Connector Loops

- Embedded connector in Lookup or Iterator mode
- Loops once for each entry returned



This allows you to do a lookup in the middle of an AL and then handle each returned entry separately.
It can be used to implement things like lookups on flat files, but be careful: such things do not scale up well!

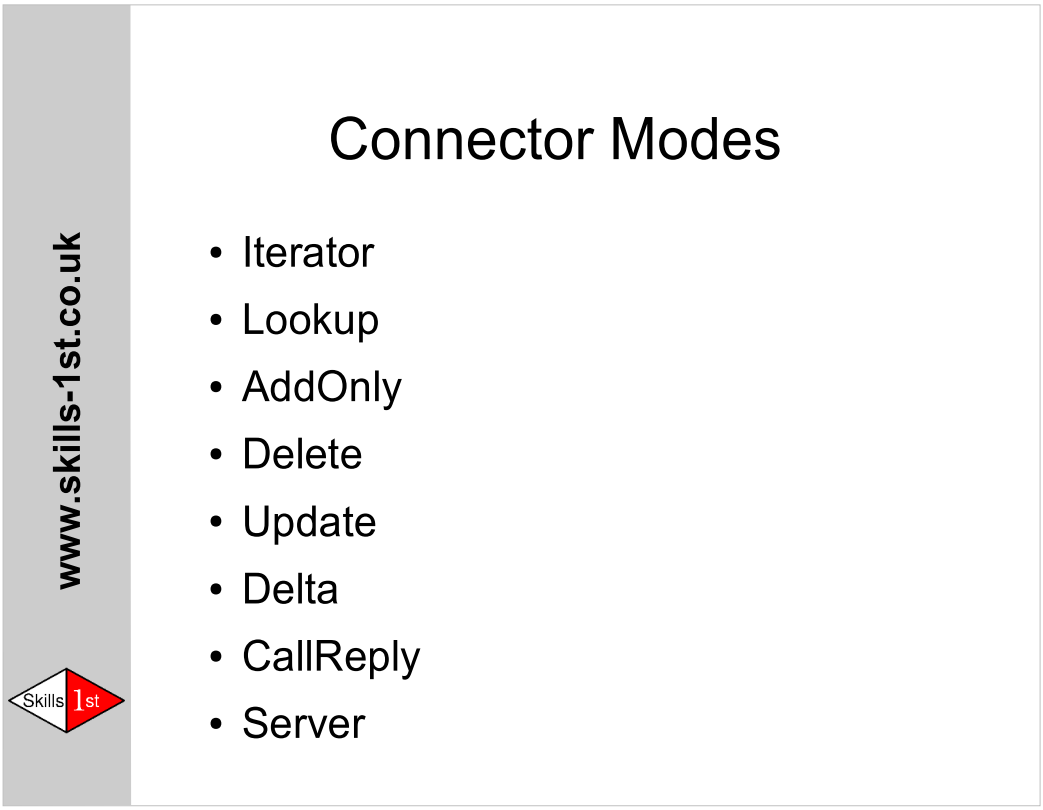
Advanced usage:
Look at the *Connection Parameters* tab: this allows you to map data from the work object into the setup of the connector – perhaps to read different files depending on some attribute value.
Note that if you do this you will probably have to change the *Initialisation* setting to cause the connector to be re-initialised for each work object.

www.skills-1st.co.uk

Skills 1st

Exercise 10

- Connector loops
- Do lookup in XML file



Every connector can operate in at least one of these modes. Some modes only exist in one or two connectors.
The modes are described in more detail in Chapter 1 of the Users Guide.

Iterator Mode

- Read every entry in a data source
- Commonly used to process files
- Some (e.g. LDAP, JDBC) can select a subset of entries by configuration
- Normally used in the Feed section

Iterators with Delta Engine

- Detect changes in data sources
- Stores copy in TDI system store
- Issues tagged attributes in work object
  - Add
  - Delete
  - Modify
- Some connectors can use these tags (Delta Mode)
  - Efficient update with no pre-read

When Delta is enabled on an iterator, it still reads every entry in the source but it only passes on those entries that have changed since it last ran.
Each attribute is tagged to show the sort of change that has occurred.
The connector stores a complete copy of the source data in the TDI System Store to make this possible.
The Delta feature is part of Iterator mode: it is not the same as the Delta Mode of a connector.
Connectors in Update mode with Compute Changes enabled will use Delta tags in preference to the Compute Changes logic.

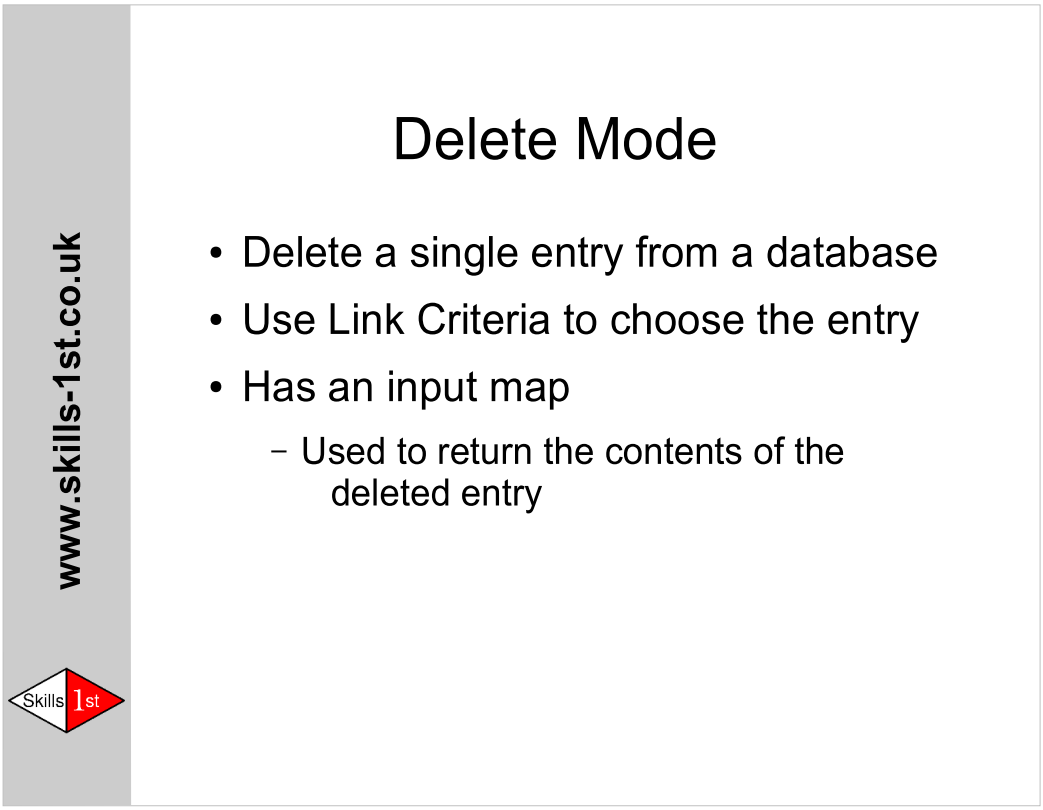
www.skills-1st.co.uk

Lookup Mode

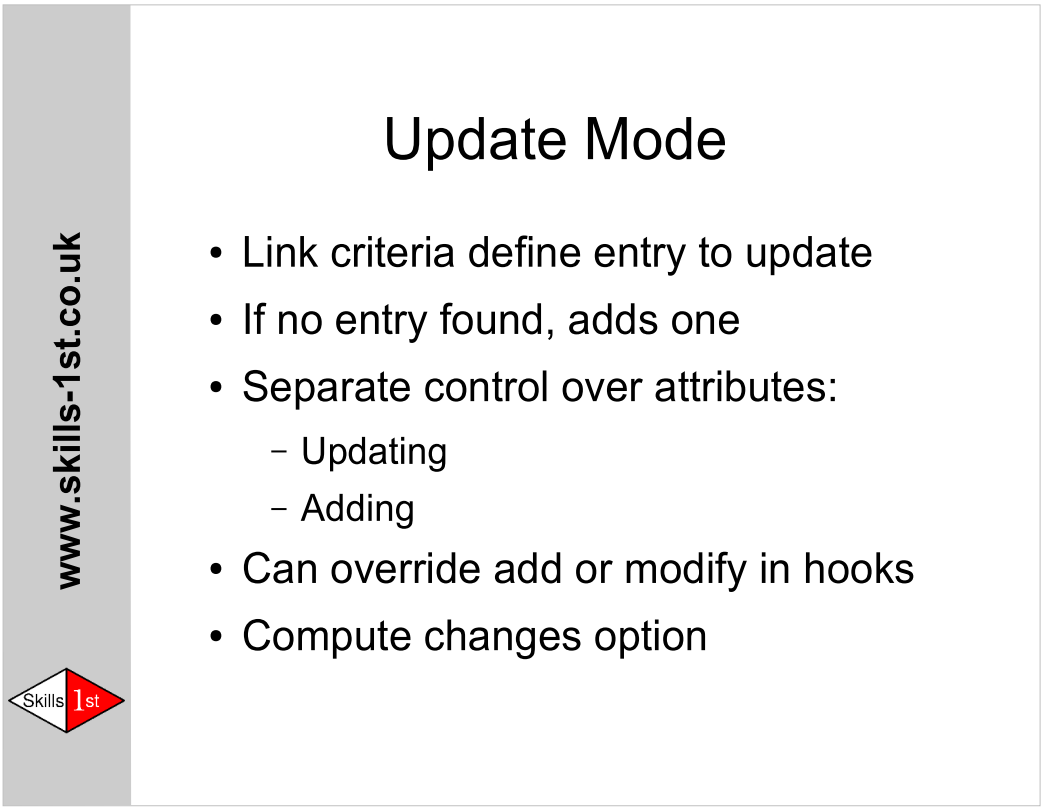
- Use attributes from the *work* object to search a data source: Link Criteria
- Import values from the data source into *work* attributes: Input Map
- Source is usually a database but can be almost anything

Add-Only Mode

- Adds one entry to a data source
- Often used to write files
- Output map only



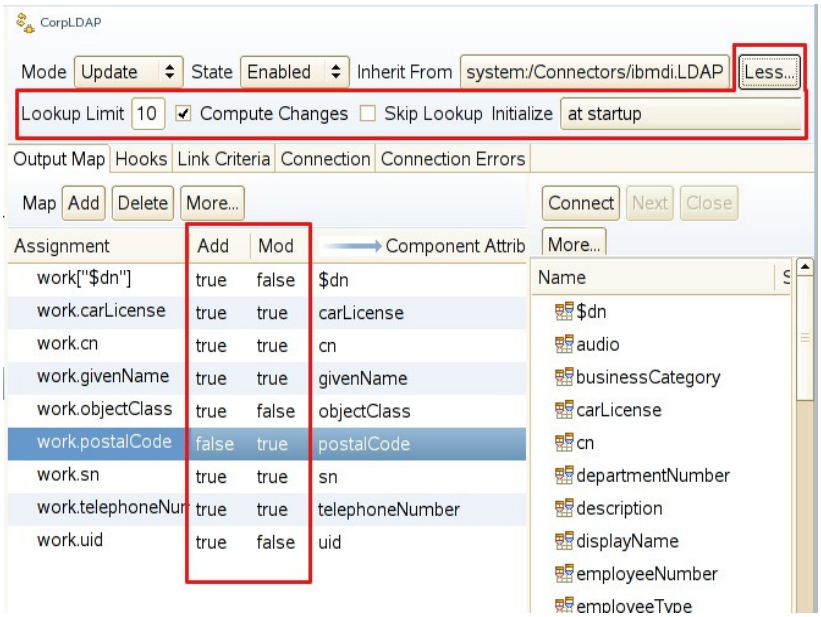
Link criteria that select no entries or multiple entries will normally result in an error, though it is possible to ignore these: be careful!



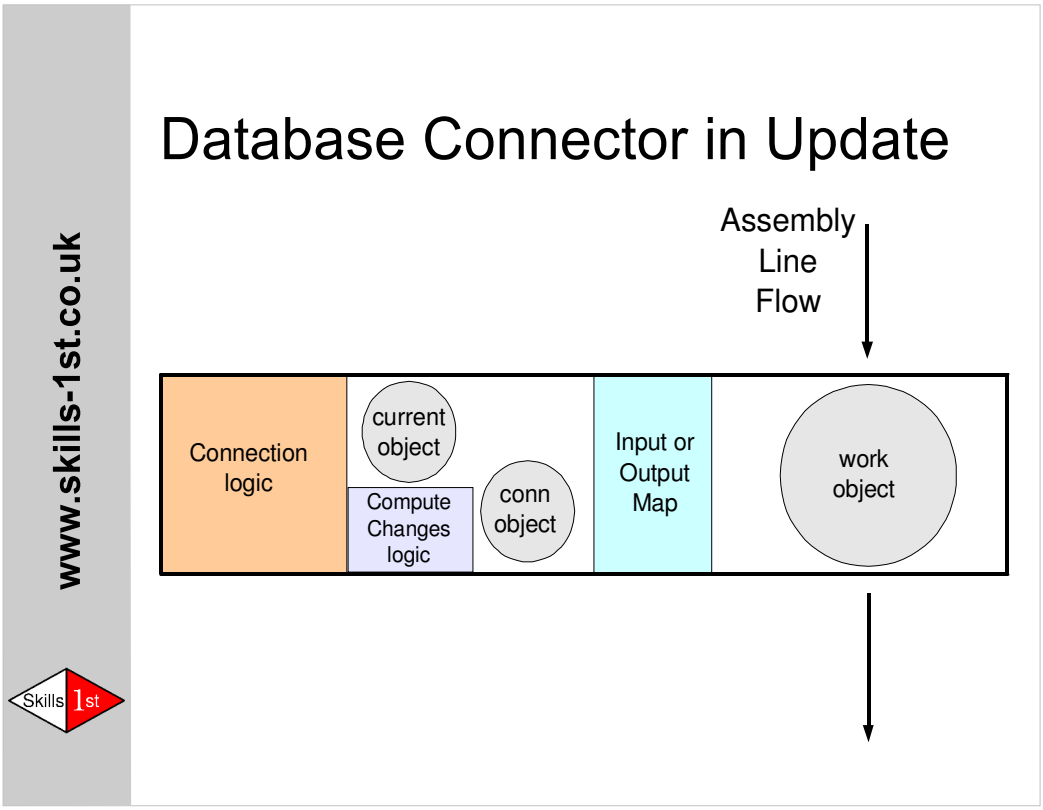
A very powerful mode – often better than Add-Only as it can maintain the data after adding it, and can also help you to create idempotent ALs.
The Compute Changes option causes TDI to read the existing entry and compare it with the desired entry (represented by the *conn* object). TDI can then issue more efficient update instructions to the database.
If Delta tags are found in an entry then they take precedence over the Compute Changes logic.

www.skills-1st.co.uk

Update Mode



Use the *More* button to access functions like Compute Changes. Each attribute can be separately enabled for Adds and Modifies. This allows you to avoid overwriting data that may have been updated by another process.



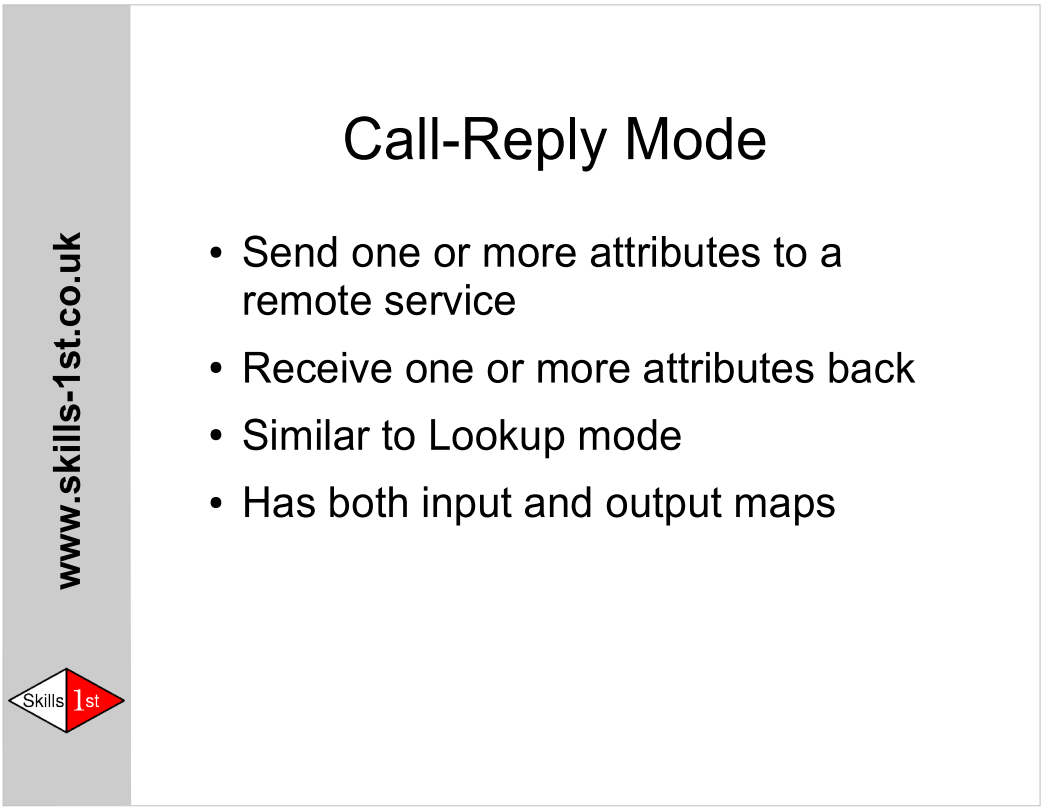
When the Compute Changes option is in effect, the *current* object is used to hold the current state of the entry selected by the link criteria so that it can be compared with the *conn* object.

www.skills-1st.co.uk

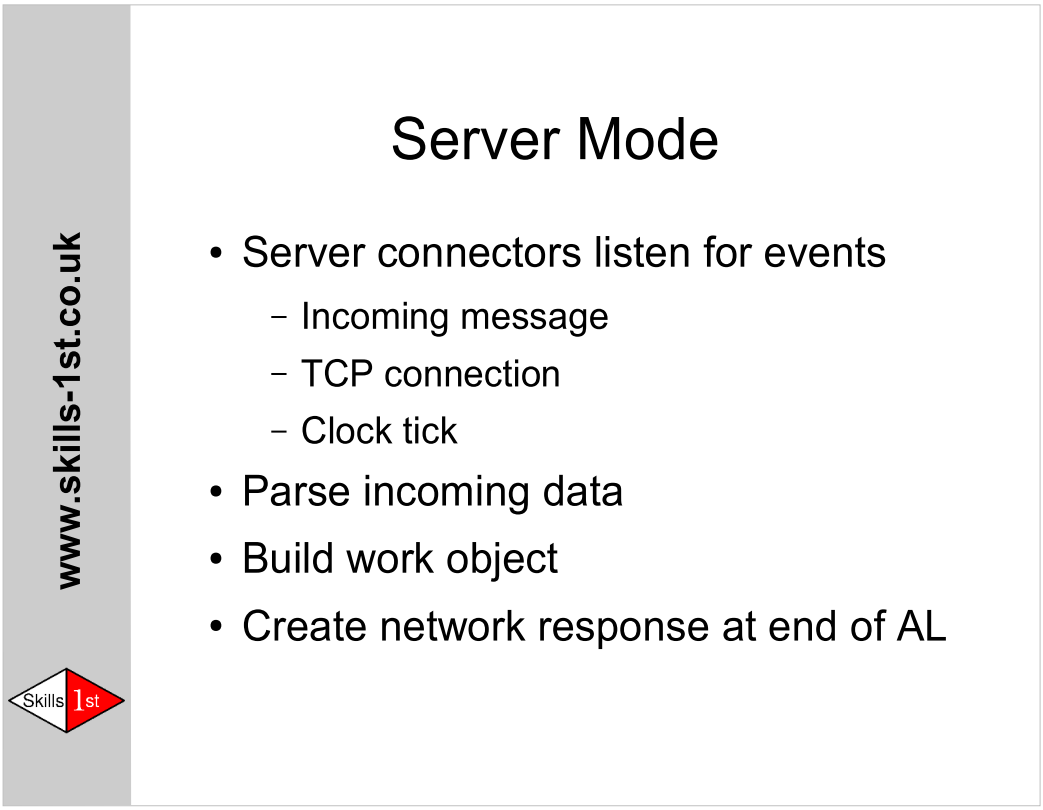
Delta Mode

- Like Update mode but uses Delta tags
- Can also process deletes
- Very efficient synchronisation when used with the Delta Engine
  - Take care! This only works if it is the *only* process updating the target data.

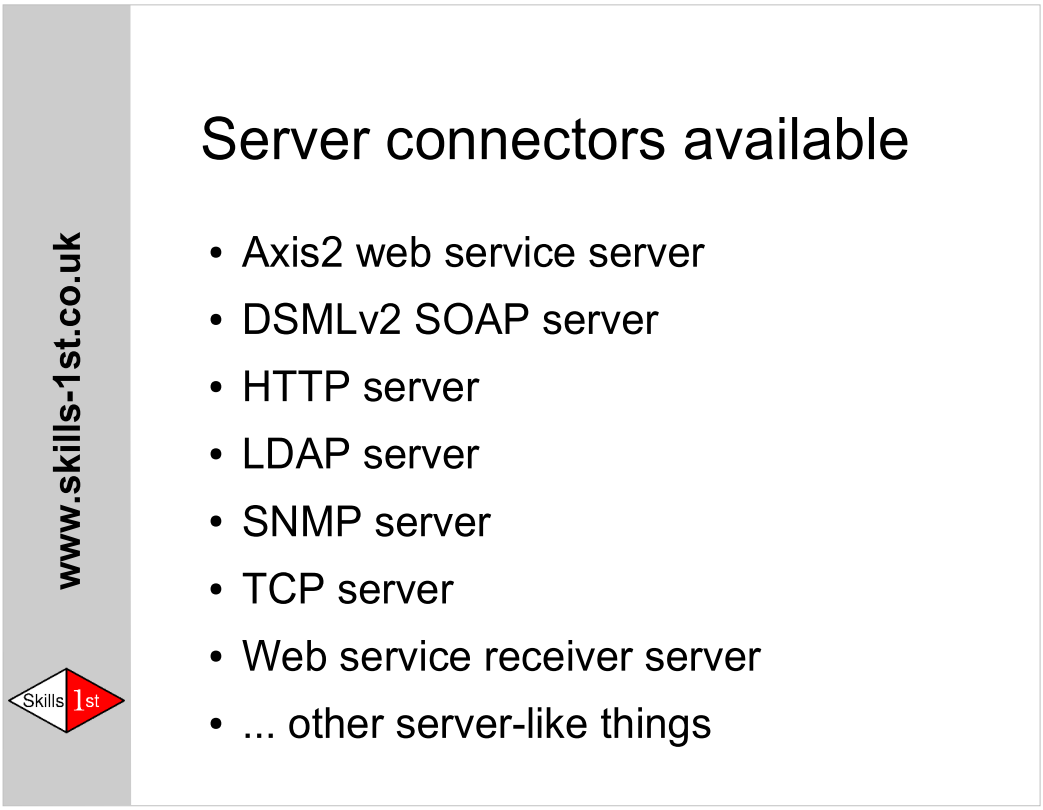
See the section on Deltas in the Users Guide.



Used where Lookup mode is not an adequate model.
Often available on messaging service connectors.



Until TDI 6.0 this feature was provided by Event Handlers. From TDI 7.0 onwards, the event handler code has been removed.



Most server connectors will only operate in Server mode, though a few will operate as Iterators.

www.skills-1st.co.uk

Skills 1st

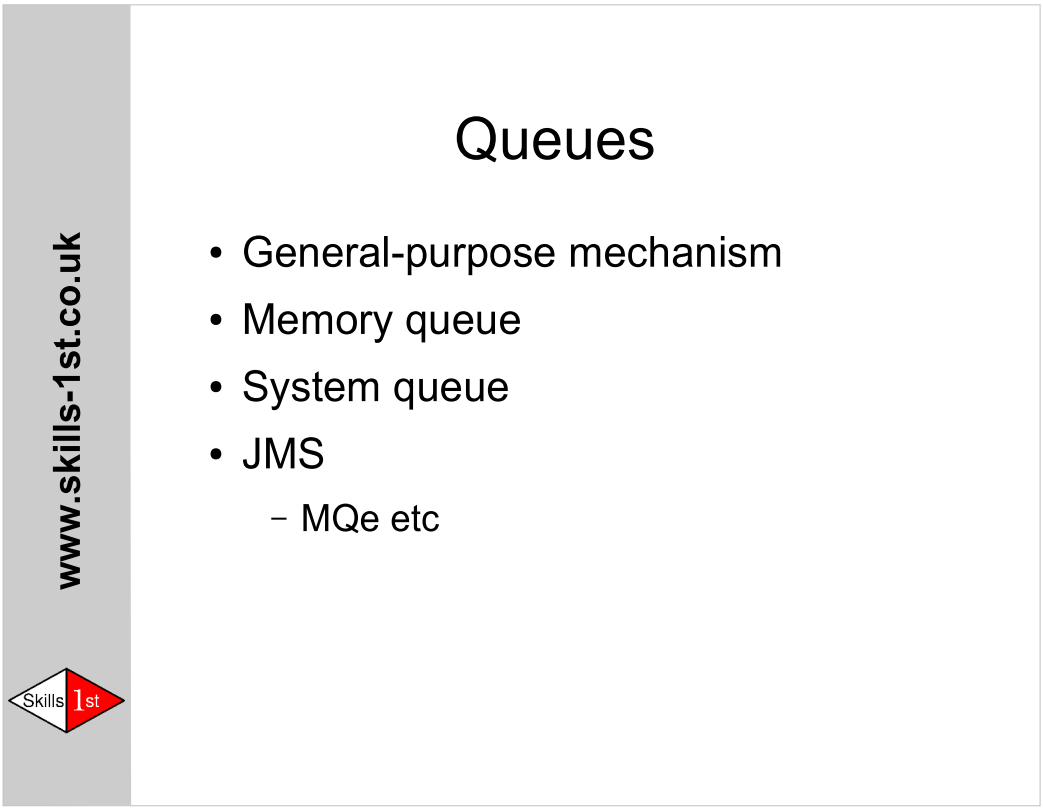
Exercise 11

- LDAP server
- Apache Directory Studio
- LDAP connector

TDI system store

- Derby / Cloudscape database
- Embedded or Server mode
  - Embedded only usable by single process
  - Server mode now preferred
- Supports Delta Engine and other features
- Can be used from scripts

Up to at least TDI 7.1 the default storage location for the data is only usable by root so you will need to change the URL in the property com.ibm.di.store.database



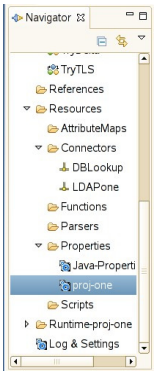
Memory queues are lost when the process exits, but most others are persistent.
Some queue types can reach remote machines.
MQe is installed along with TDI, but you may have to configure it.
A queue called '_default' should have been set up for you by the CE when the project was created: if you only need one queue then this may be adequate.
Note that MQe is the default system queue provider, but that it is deprecated in TDI 7.1. No successor technology has been nominated, but ActiveMQ looks to be a likely candidate.

www.skills-1st.co.uk

Skills 1st

Property Stores

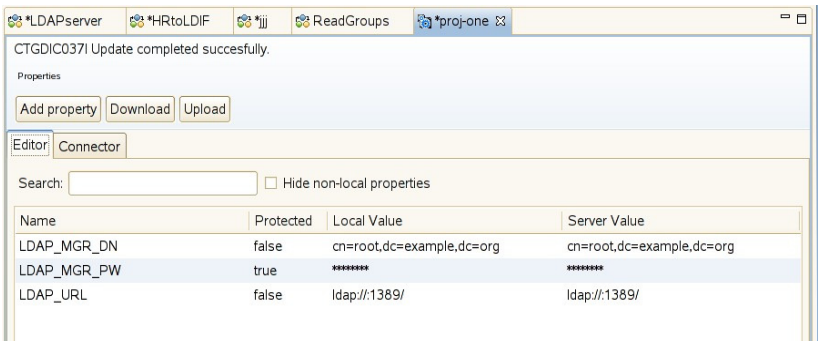
- Remove connection details from config
- Name-Value stores
- Multiple stores possible
- CE can upload to remote stores
- Properties can be protected (encrypted in file)



Remote (and per-project) property stores are implemented in the TDI System Store.
Remember that the runtime can decrypt encrypted values, so the protection will not stand up against a moderately determined attacker.

www.skills-1st.co.uk

Edit Properties



See *Connector* tab for file location etc.
Most configuration items in TDI can take values from properties: click on the item name or the '?' button beside the value field.
You can access properties from scripts: system.getTDIProperty(propname)

www.skills-1st.co.uk

Skills 1st

Exercise 12

- Build a property store
- Use properties in LDAP configuration

Handling variable-size results

- Zero results may not be an error
- Multiple results may not be an error
  - OK to have multiple phone numbers
- Handle these cases with loops:
  - Connector loops when searching
  - Attribute-value loops when updating
- Scripts can add new values to attributes

Lookups can return any number of results.
In versions of TDI before 6.0 these cases had to be handled entirely by scripting. Loops allow a more visible solution.

www.skills-1st.co.uk

Building Attributes

- work.newAttribute(“telephoneNumber”)
- work.telephoneNumber.addValue(...)
- work.getString(“phone”)

www.skills-1st.co.uk

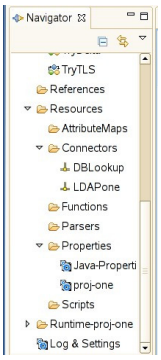
Skills 1st

Exercises 13 and 14

- Build multi-value attributes
- Build a web-server that answers white-pages queries

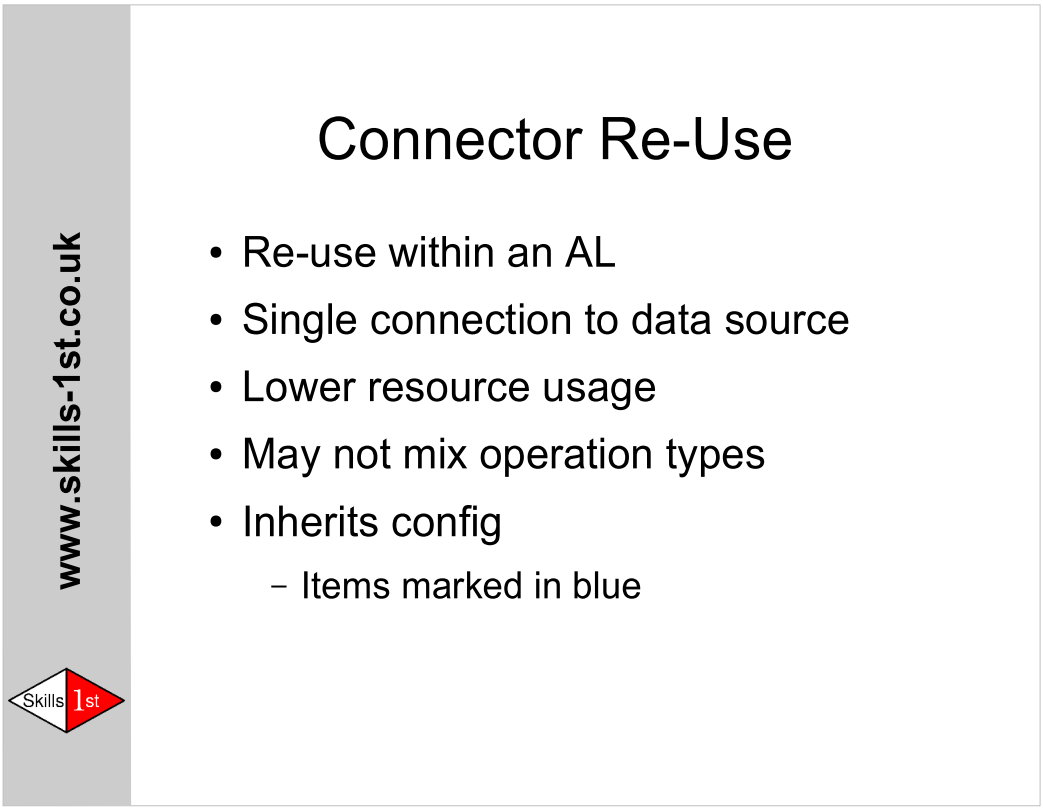
www.skills-1st.co.uk

Building a Library

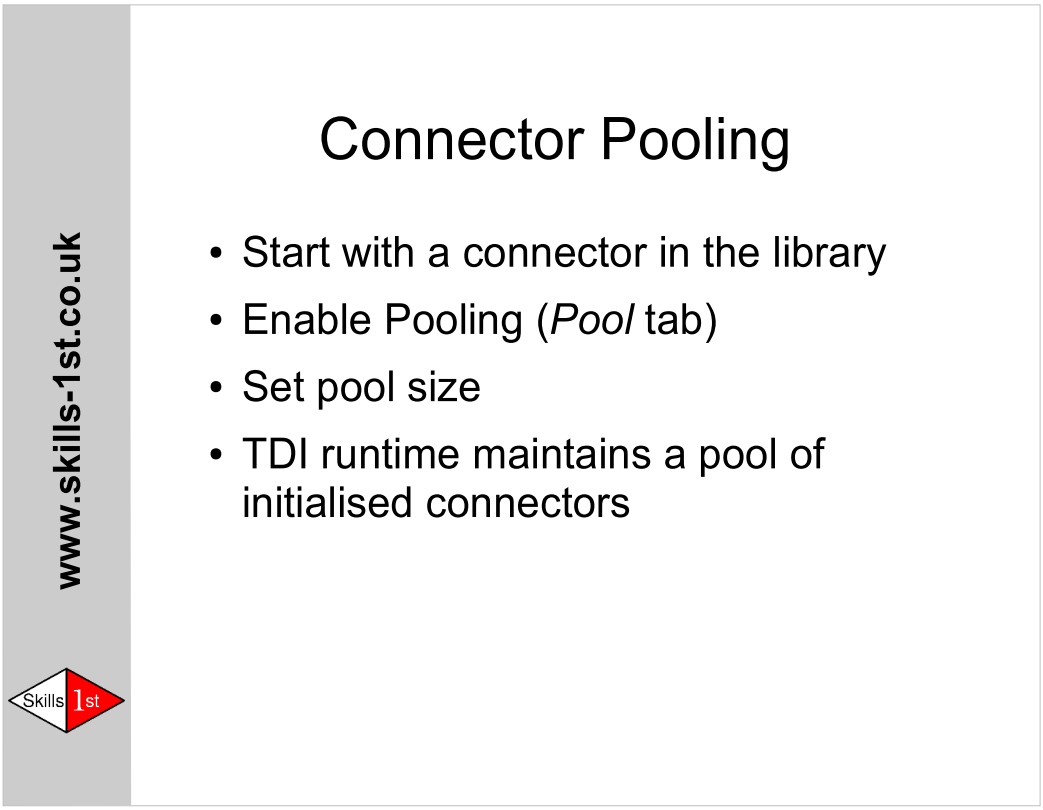


- Resources
  - Attribute Maps
  - Connectors
  - Functions
  - Parsers
  - Properties
  - Scripts
- Inheritance

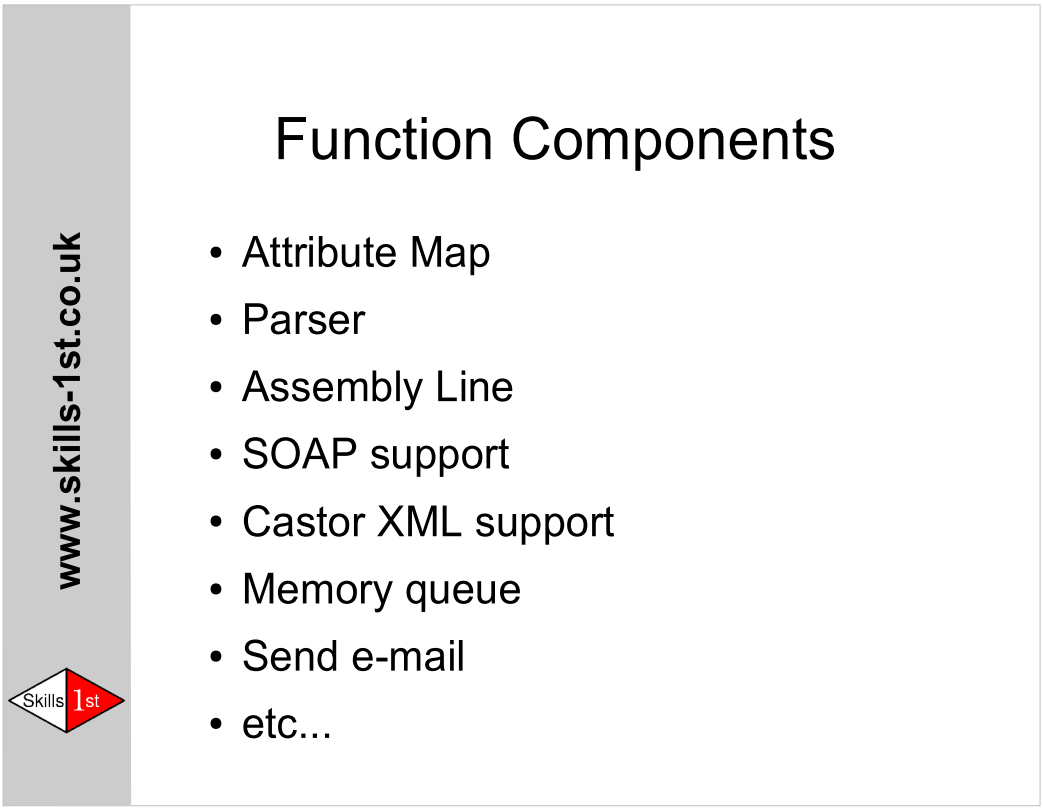
Most projects involve several assembly lines connecting to the same set of data stores. Using a connector library avoids having to repeat the configuration work each time.
When you use things from the library they inherit much of their configuration from the library copy: this makes it easier to do system-wide changes.



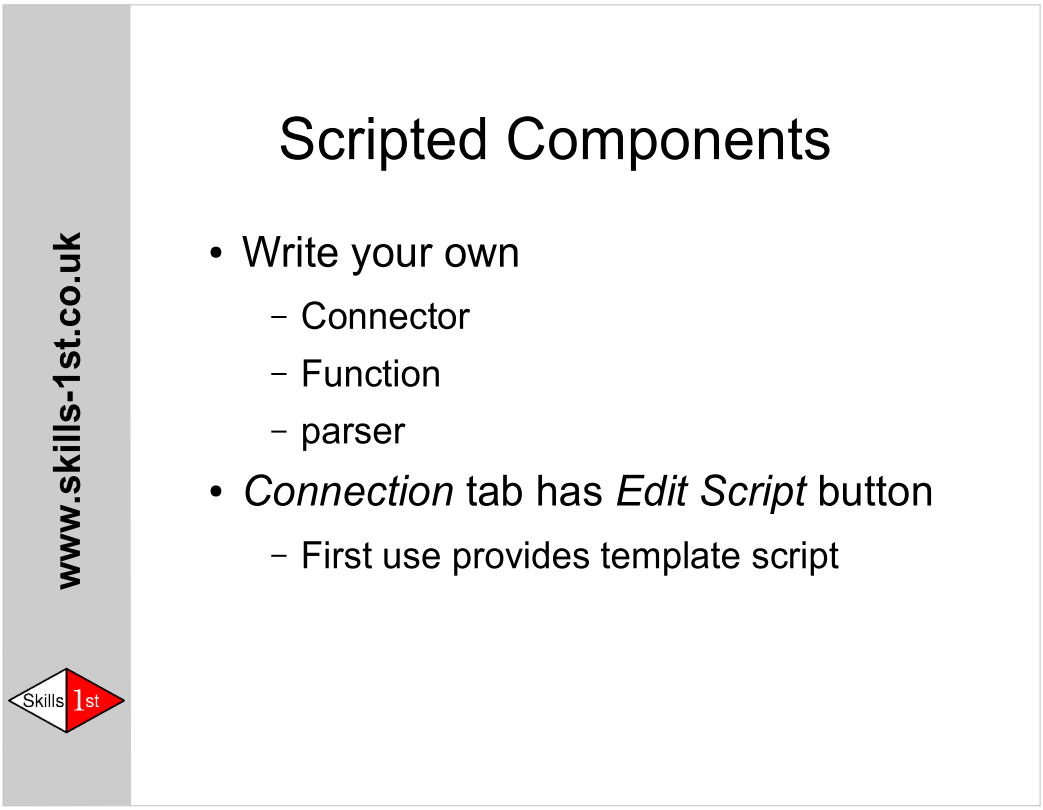
Not the same as using a connector from the library: this is using the same instance of a connector multiple times within one AL.



Useful when building high-performance systems.



Some of these look a bit like connectors (Assembly Line, Memory Queue, send e-mail) but most act on data within the work object. The Delta Engine is available as a function component so you can embed it at any convenient point in the AL.

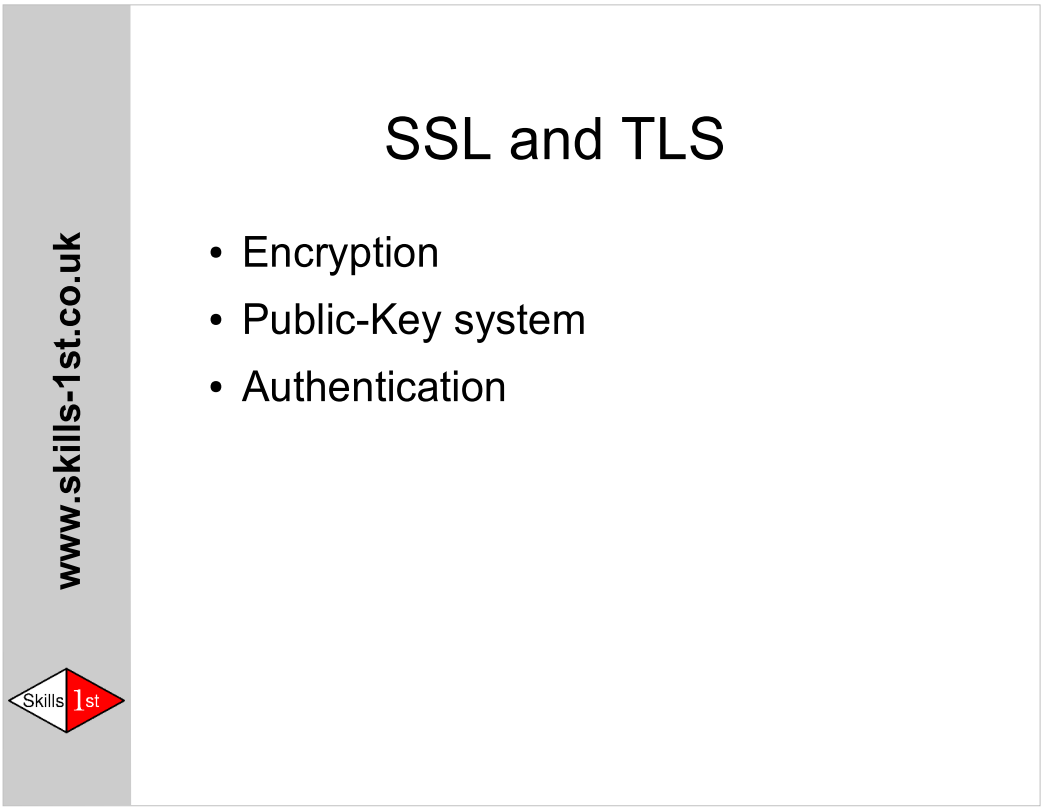


Something missing from TDI? Just build it!

www.skills-1st.co.uk

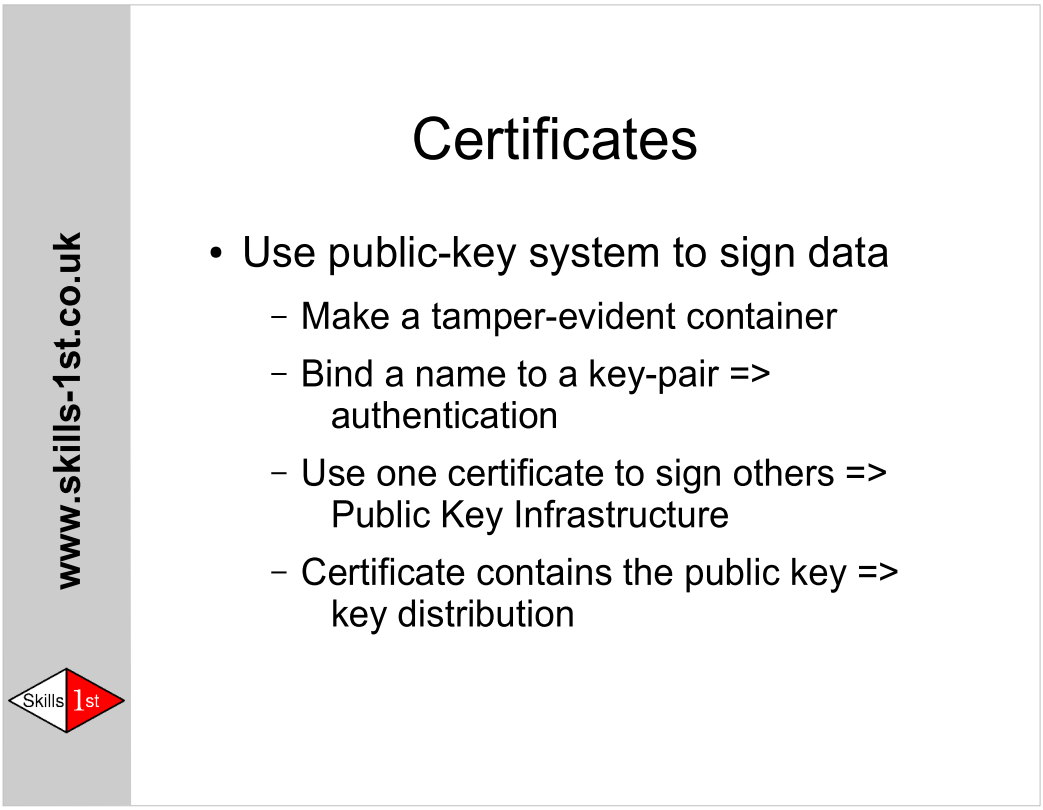
Exercise 15

- Delta mode

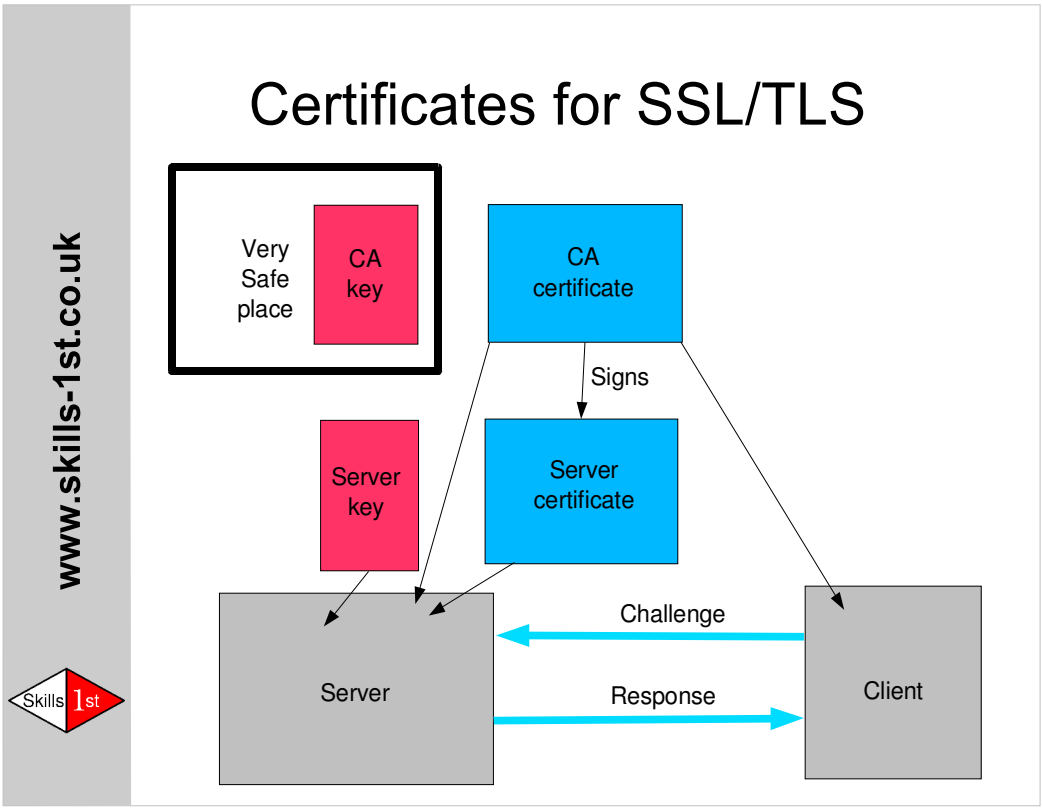


Encryption is an essential part of the security of most IT systems.

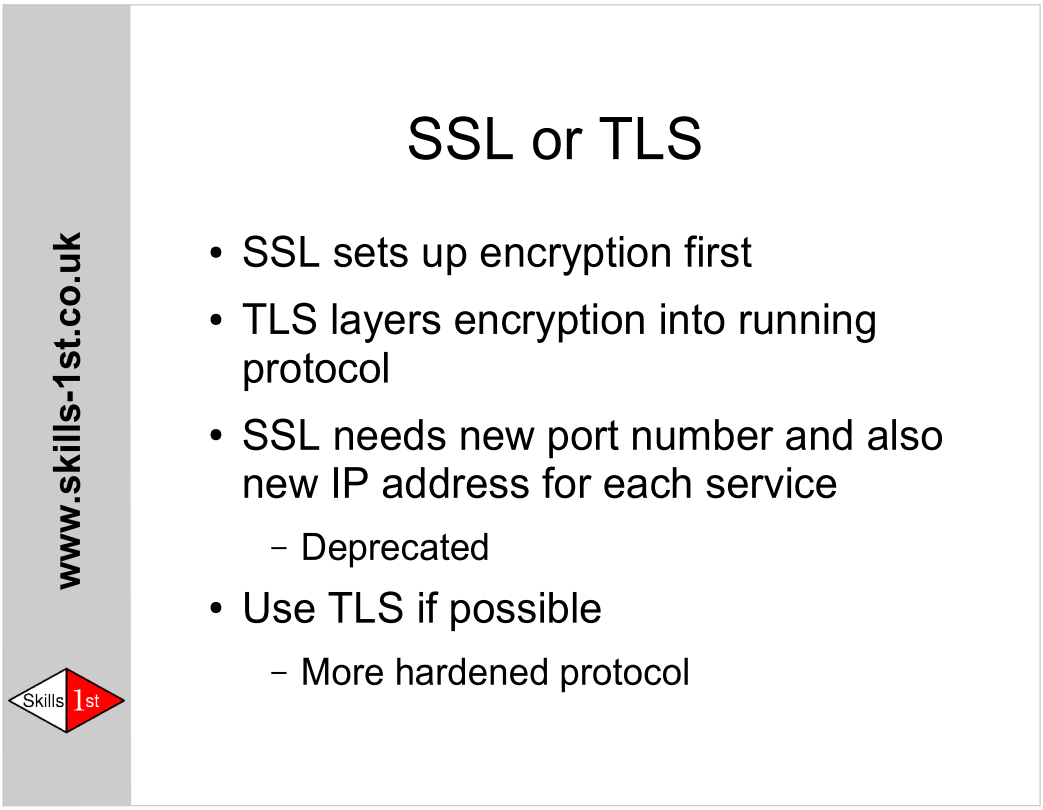
Public Key cryptosystems allow anyone to create a message that can only be read by a particular recipient. This vastly simplifies the key-management problem.
A zero-knowledge proof is a scheme that allows me to prove that I know a particular secret to someone who does not know that secret. It is used for secure authentication, as the entity being authenticated never has to disclose anything that could be used by someone else to impersonate them.



The most common certificate format used today was standardised as part of the X.500 directory services definition.



Usually the client challenges the server to prove its identity by signing a random number supplied by the client.
Optionally the server can challenge the client in the same way, to provide very secure mutual authentication. The client needs a key and certificate of its own for this to work.



The current version of TLS is 1.2 which is defined in RFC5246 (August 2008). This completely superseeds the old SSL protocols, which are all now vulnerable to known attacks.
Even TLS 1.2 has a flaw, which was discovered in August 2009 and allows a specific form of man-in-the-middle content-injection attack. This is fixed by RFC5746, but some implementations may not have caught up yet.

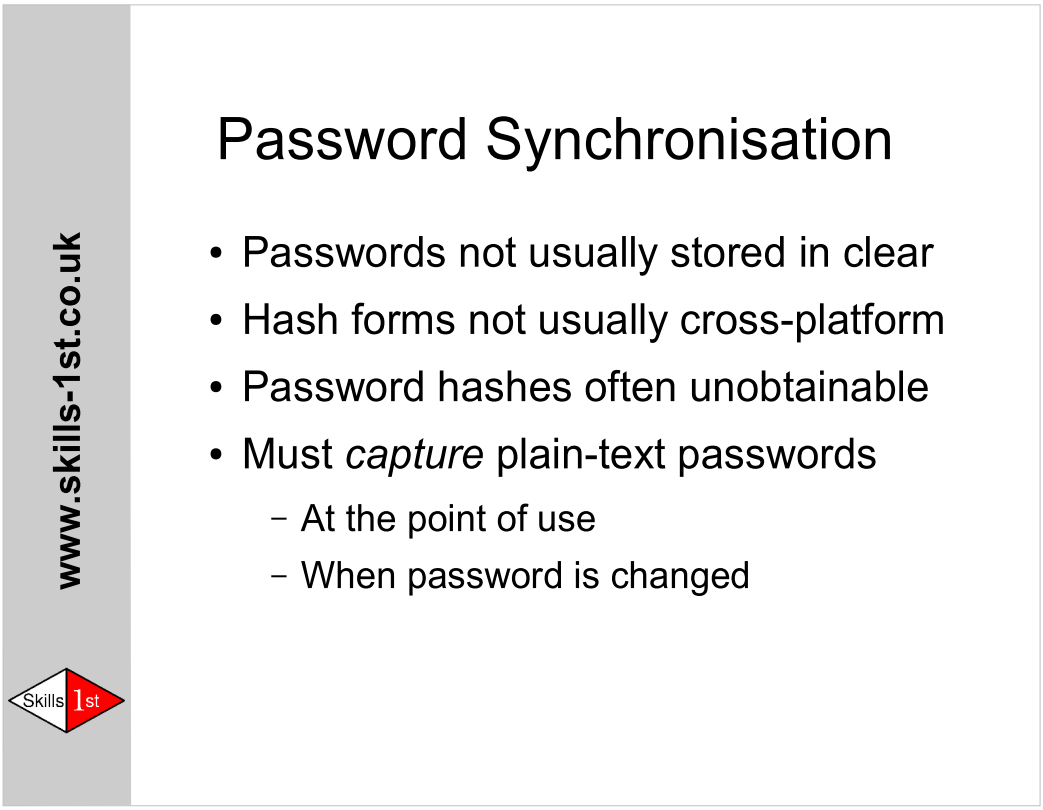
www.skills-1st.co.uk

Skills 1st

Exercise 16

- Use TLS with LDAP

TDI supports SSL out of the box, but TLS is the better protocol.



If you are very lucky your systems will all share a common password hashing algorithm. Many LDAP servers can work with several different hash types, often covering all the variants used by Unix and Linux.

Password Capture Plugins

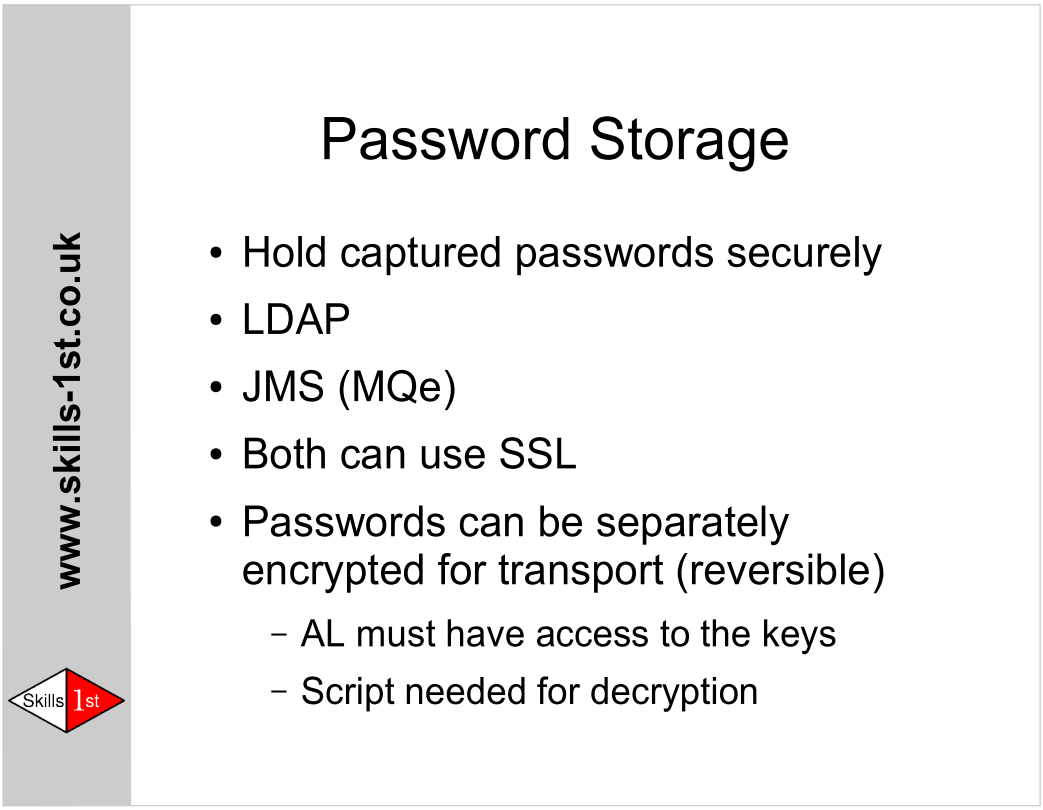
- TDI supplies plugins for
  - Standalone Windows (XP and later)
  - Active Directory domains
  - Sun Directory Server
  - IBM Tivoli Directory Server
  - Lotus Domino
  - Unix/Linux PAM systems

Example Plugins

- Active Directory
  - DLL
  - Acts as a password-quality checker
  - Must be installed on all domain controllers
- Unix/Linux PAM
  - Shared Library
  - Configure into the normal PAM stack

Alternatives to the plugins

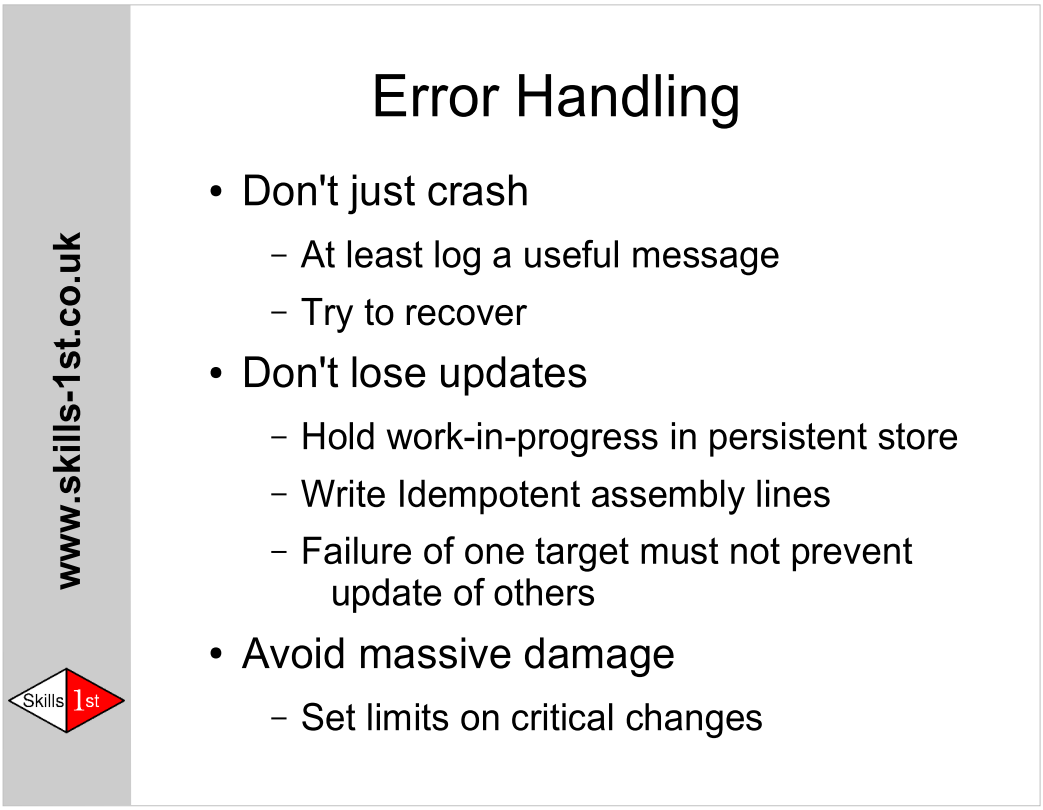
- Ordinary LDAP searches
  - If passwords stored in clear
  - If password hashes are usable
- Other capture mechanisms
  - Microsoft Services For Unix (SFU)
  - Password Hook: passwdhk.sourceforge.net



Passwords are extremely sensitive data!
Try not to store any form that can be converted to plain text.
If you *have* to store passwords in transit, make sure that they are deleted securely (this is hard).

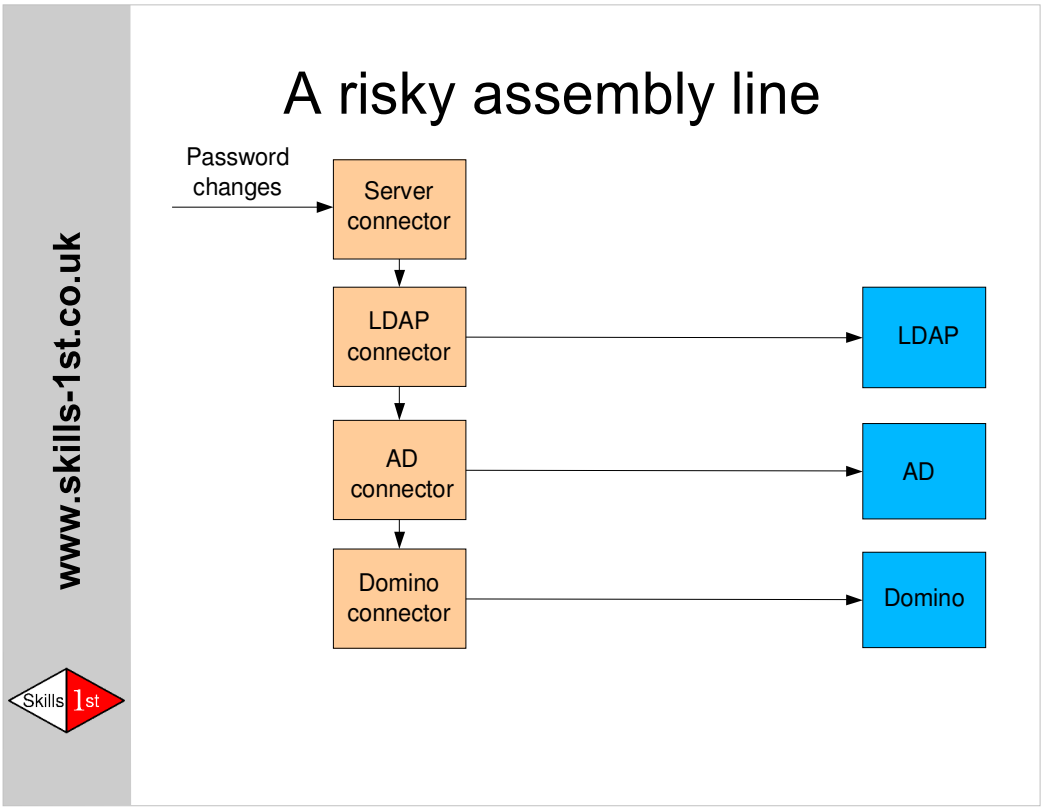
Robust Systems

- Good error-handling
- Not losing updates on error or crash
- Recover from connectivity problems
- Testable components
- Good architecture
  - Simple single-task ALs
  - Each data flow independent of others
  - Well-defined, replaceable ALs

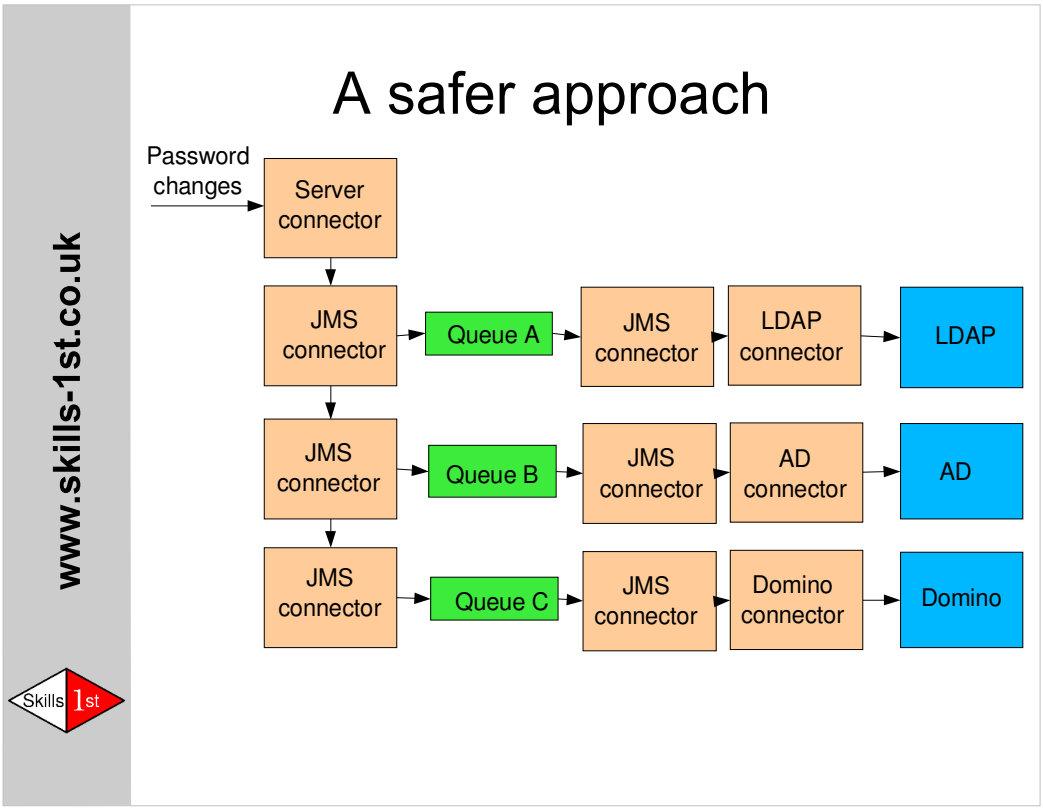


Idempotent: Describing an action which, when performed multiple times, has no further effect on its subject after the first time it is performed. Thus an Idempotent AL can safely be restarted if it fails.

In Assembly Line Settings, can set limit on number of errors. May want to count delete or modify ops and stop if there are too many in a short space of time.



Can you see the problem here?
There are actually several different failure modes.



More complex but much safer.
There are still ways in which this can fail, but they are much less likely than in the previous example.

Reconnecting

- *Connection errors* tab in connector
- Retry on:
  - Initial connection failure
  - Connection loss in operation
- Configure timeout and retry limit
- Beware: takes time, blocks error hooks

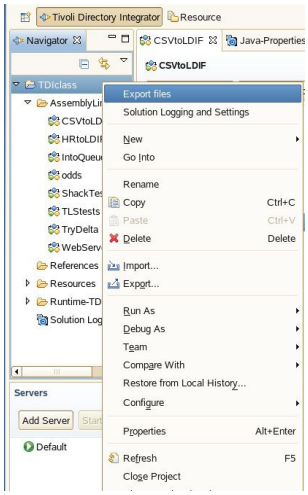
Going Into Production

- Export config files for runtime
- Property files
- Boot-time startup
- Scheduled tasks
- Logging
- Monitoring
  - AMC
  - Action Manager

www.skills-1st.co.uk

Exporting the runtime config

- Right-click on the project
- Select *Export Files*
- Choose a directory
- Choose ALs to run



You will normally get several files:
- Runtime config *projectname.xml*
- Project-specific properties files
- Startup script

You may need to copy other files to the production environment:
- Global properties
- Solution properties
- Encryption key files

Starting the runtime server

- Installer adds lines to /etc/inittab on Unix-like systems: server runs as *root*
- Installer creates a system service on Windows: server runs as *System*
- Consider security!
  - Does the task *need* that much power?
- Consider command-line startup
  - ibmdisrv
  - tdisrvctl

Read Chapter 6 (security) of the Admin Guide.
Read Chapter 13 (command-line options).

Scheduling ALs

- Timer Connector
  - Feed a work object into an AL at certain times of day
  - Scheduling AL could call others
- Crontab
  - Invoke ALs from command-line

Monitoring TDI

- Build an AL to monitor the others
  - Report to enterprise console etc.
- AMC: Admin and Monitoring Console
  - Bundled with TDI
  - Can invoke “Health AL” and display result
- Action Manager
  - Automatic actions based on rules
- SNMP

www.skills-1st.co.uk

Exercises 17 - 23

- Prepare for production
- AMC
- Report differences between data sources
- Account provisioning

www.skills-1st.co.uk

Summary

- TDI structure
  - CE and Runtime
  - Assembly Lines, Connectors, etc
  - Scripts and hooks
  - Debugger
- Solution design

www.skills-1st.co.uk

What have we missed?

- Many advanced topics
  - Creating re-usable TDI components
  - Access-control for runtime API
- Many types of connector
- Many types of parser
- Action Manager

www.skills-1st.co.uk

Learning more

- TDI Users website:
  www.tdi-users.org
- Google Group:
  ibm.software.network.directory-integrator
- PDF manuals from IBM website