

## Introduction

These guidelines provide context and guidance to customers to integrate Talis Aspire, a resource list management system, into their institutional VLE.

## Background

### What is Talis Aspire

Talis Aspire is a hosted resource list management system supplied by Talis Education Ltd. It allows academics to collect resources from their library catalogue as well as the wider web, and then build resource lists efficiently using a simple drag and drop interface. With Talis Aspire, the library is included within the overall workflow, helping to ensure timely purchase of appropriate stock, thereby meeting student expectations of easy access to listed resources.

### Making Aspire more discoverable to users

For the majority of institutions adopting Talis Aspire, the VLE is the primary tool for exposing the resource list to students, and is the first place students will look to find their lists.

To ensure this user expectation is met, and to maximise the return on investment an institution has made in Talis Aspire and its VLE, an integral part of any Talis Aspire implementation project is to ensure that academics can easily place their resource list into the VLE for student access.

*Note: The responsibility for integrating Talis Aspire (e.g. into the VLE) lies with the institution, although Talis will be available to provide support and assistance as required.*

### Aspire integration pre-requisites

We recommend completing the following as a minimum within your Aspire implementation before commencing integration work:

- Tenancy setup: Setting up and configuring your institution's own Talis Aspire system.
- Authentication: Integration of your Talis Aspire service with your local authentication mechanism (e.g. Shibboleth; OpenAthens).
- Hierarchy conversion: Configuration of Talis Aspire to your institutional structure or hierarchy.
- Data Conversion: Either a completed first-cut data conversion or, through the delivery of a working authentication system, successful manual input of several test resource lists.

### The importance of the hierarchy

As part of your project implementation, Talis Aspire will be configured to a representation of the institution's structure or hierarchy. This information can be provided from various sources and describes not just the entities themselves (e.g. schools; departments; courses; modules), but also the relationships between them.

An occurrence of any one of these entities is referred to as a *node* within Talis Aspire, and contains some basic information which includes the node name, identifier, description and associated lists. As

will be demonstrated, both the node type (e.g. programme, module) and the node identifier (e.g. ABF203) are critical in providing different discovery paths to Talis Aspire.

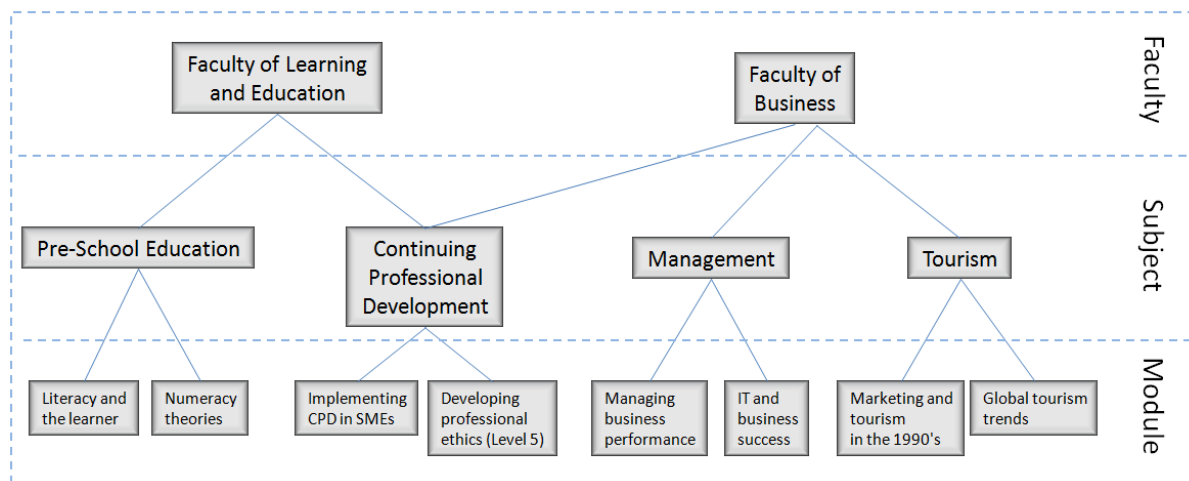


Figure 1: A simple snapshot of an example hierarchy

## Approach

The primary requirement is the provision of a link in the VLE to the corresponding page within Talis Aspire – typically, a link from a VLE module page to that same module node in Talis Aspire.

### Integration using the “linking” API

The node URLs within Talis Aspire follow a consistent construction. This means a developer can programmatically (i.e. logically) determine what the URL is to automatically insert it within the VLE. This avoids both the effort of manually inserting URLs to Talis Aspire into the VLE, as well as the ongoing effort involved in maintaining them. This construction is shown in Figure 2:

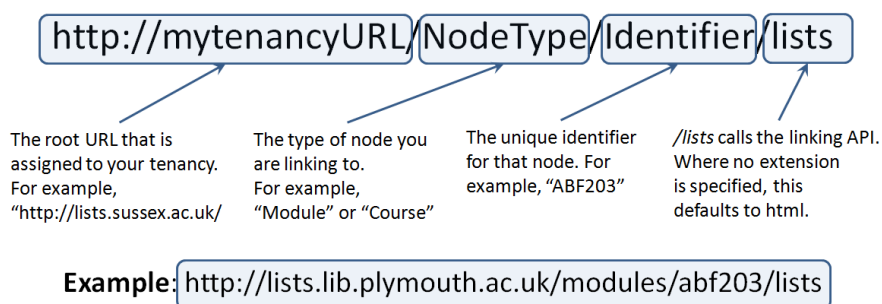


Figure 2: Construction of a URL using the linking API

The most basic integration is simply inserting this URL as an HTML link into the VLE. There are four possible scenarios that result from a user following a link of this construction:

**Scenario One:** A corresponding node exists in Talis Aspire, and only a *single list* is associated with this node.



Figure 3: User clicks linking URL where only single list associated with node

**Scenario Two:** A corresponding node exists in Talis Aspire, and *two or more lists* are associated with this node.

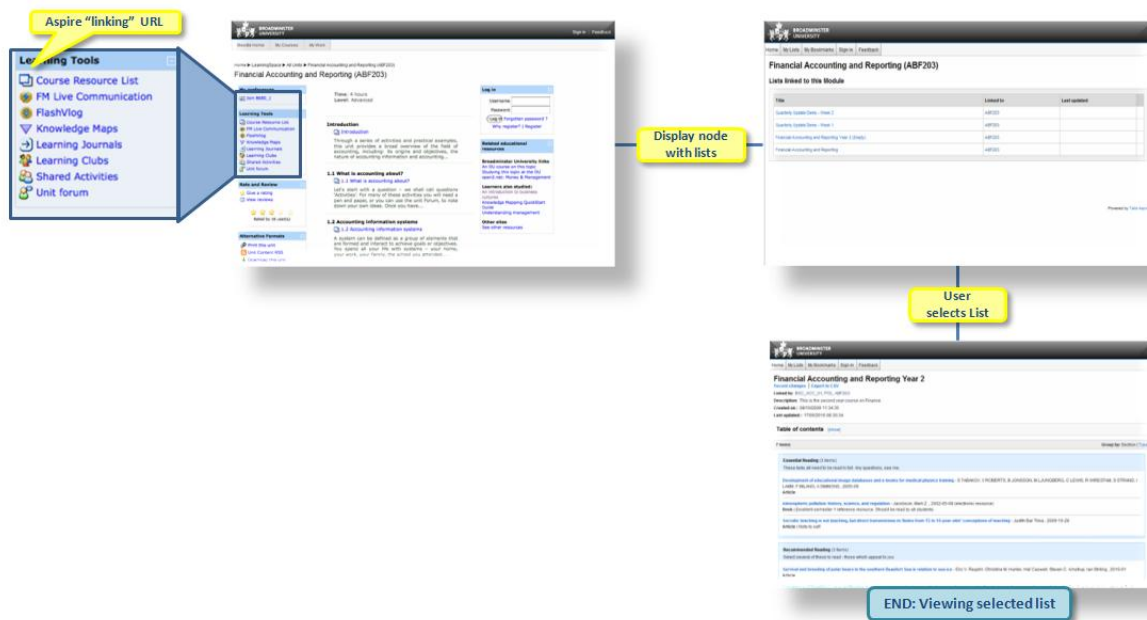


Figure 4: User clicks linking URL where multiple lists associated with node

**Scenario Three:** Although a matching node exists in Talis Aspire, that page has no lists associated to it (see Figure 4).

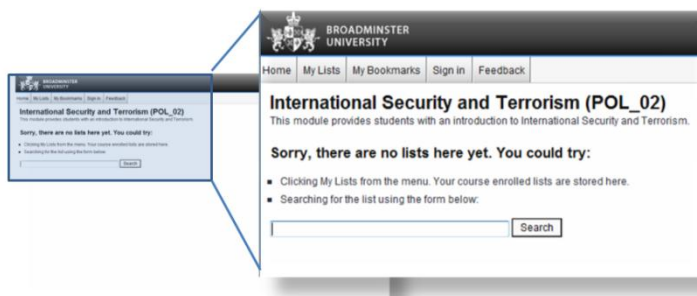
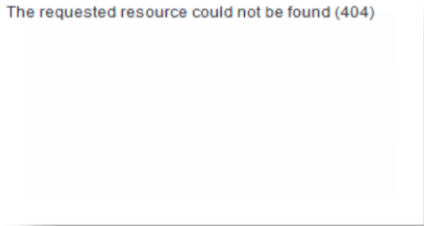


Figure 5: Search prompt provided where the node exists but no lists associated

**Scenario Four:** No node exists within Talis Aspire for that URL. The browser returns a standard HTML error page “404 Error - File Not Found” (See Figure 3).



The requested resource could not be found (404)

Figure 6: Returned 404 Error where no node exists

### Problems with the basic HTML approach

Used in HTML mode, the linking API provides a more consistent user journey from the VLE to Talis Aspire. However, this approach can fail to optimise the quality of the user’s experience.

- Where no node exists in Talis Aspire for the generated URL, the user is left to “cope” with a 404 Error page
- The user needs to click the link in the VLE to determine if a list exists.

Such exceptions should ideally be dealt with in the VLE itself.

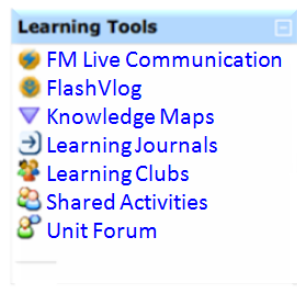
### Advanced use of the Linking API

It is possible to address these issues by using the linking API and requesting alternative data representations of the node page (see over JSON/RDF). This approach requires some programming expertise within the institution, usually from within the VLE team itself, but can yield a far more successful integration.

By way of example, compare a typically-constructed link with two scenarios that have made use of the linking API.



- A typically constructed link. Users taking this link have no on-screen feedback as to what to expect on arriving at their destination.



- A programmer has used the linking API to determine that either (a) the module doesn’t exist in Talis Aspire for the module being linked from or (b) there are no lists on the module. A link, that might otherwise mislead the user, is not displayed. This example could be extended further, for example prompting the student to contact their module tutor for more information rather than simply displaying no list.



- A programmer has used the linking API to bring back information on the lists associated with the node (titles and resource count). This is a far richer experience.

The linking API can return two other data representations, which a programmer can use to refine the display properties on the VLE link to Talis Aspire:

**JSON:** JavaScript Object Notation is a lightweight computer data interchange format. It is a text-based, human-readable format for representing simple data structures and objects.

Example: <http://mytenancyURL/NodeType/Identifier/lists.json>

**RDF:** Resource Description Framework is a metadata framework that provides interoperability between applications that exchange machine understandable data on the Web.

Example: <http://mytenancyURL/NodeType/Identifier/lists.rdf>

Should you wish to learn more about using these RDF/JSON representations to improve the quality of your users' experience, please contact your Talis representative who will be pleased to help.

### A final consideration

As has been shown, it is possible to build a rich integration between the VLE and Talis Aspire. However from our experience, we recommend taking the time to consider one simple question before commencing – how does the link get added to the page within the VLE? It is in answering this question that institutions will select the best approach to linking within the VLE. For example, you may wish to:

- **Place a link on every appropriate VLE page, regardless of whether a list exists**
- **Automatically generate a link on a VLE page, but only when a list exists**
- **Allow the list creator to say “I’ve done my list; now generate me a link that I can copy/paste where I want”**

There are many variations of these scenarios, and other scenarios besides. The institution should also consider whether the list creation / publishing is mediated by the library, or whether there is a mixed economy (i.e. some lists added by library staff with others added by academics themselves). With the latter, the approach by which the list link gets added needs to support both.

We would recommend that the library and VLE team liaise on defining the experiences of both list creators and list users. The questions that may need discussion include:

- **Who adds the link?**
- **When, in the life-cycle of the list, is the link added?**
- **How is the link added - manually or automatically?**
- **How does the link (or links) display to the user?**
- **What experience do we want to provide?**

### Advice and support

The guidelines should provide customers all the information to consider integrating Talis Aspire with the VLE. However, should you have any additional queries or require further advice, there are several options available:

- Contact your Talis Aspire Project Co-ordinator, who will either answer your queries, or arrange for either a business or technical expert to contact you
- Raise a support call
- Post your question to [lis-talis-aspire@jiscmail.ac.uk](mailto:lis-talis-aspire@jiscmail.ac.uk), an open forum for the Aspire community. Other institutions who have already integrated an equivalent VLE may be willing to share their experiences.

END