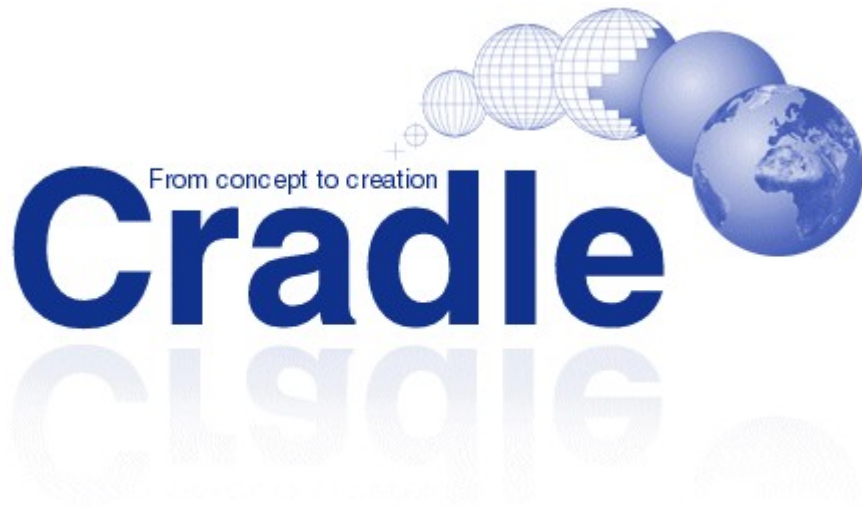


# Overview



Structured Software Systems Ltd  
Suite 2  
22a Duke Street  
Barrow in Furness  
Cumbria, LA14 1HH  
United Kingdom

Telephone: +44 (0) 1229 838867  
Fax: +44 (0) 1229 870096  
e-mail: [info@threesl.com](mailto:info@threesl.com)  
Website: <http://www.threesl.com>

Document: RM018  
Issue: 12  
Issue Date: February 2009  
Product Version: 6.2

Printed in the United Kingdom

Copyright 2010 Structured Software Systems Limited. All rights reserved.

Copyright © 1988-2010, an unpublished work by Structured Software Systems Ltd (3SL). All rights reserved. This work contains licensed material that is the property of 3SL and may only be used in accordance with the terms of a Licence Agreement with 3SL. Any unauthorised use, duplication or disclosure of this work, and any unauthorised copying or reproduction of the User Interface or other documentation that form part of this work, are prohibited and may result in prosecution. Cradle is a registered trademark of 3SL in the United Kingdom and other countries.

The software programs described in this document are the confidential information and property of  
Structured Software Systems Limited.

THE SOFTWARE PROGRAMS, EITHER ACTUAL OR PROPOSED, DESCRIBED IN THIS DOCUMENT ARE, OR ARE INTENDED TO BECOME, THE CONFIDENTIAL INFORMATION AND PROPRIETARY PRODUCTS OF STRUCTURED SOFTWARE SYSTEMS LIMITED. DISCLOSURE OF THE CONTENTS OF THIS DOCUMENT TO ANY PERSON, PERSONS, OR ORGANISATIONS IN ANY FORM WITHOUT THE PRIOR WRITTEN PERMISSION OF STRUCTURED SOFTWARE SYSTEMS LIMITED IS EXPRESSLY FORBIDDEN. THE MAKING OF ANY COPIES OF THE SOFTWARE OR OF THIS DOCUMENT BY ANY MECHANICAL, ELECTRO-MECHANICAL, PHOTOGRAPHIC, ELECTRONIC, OPTICAL, OR COMPUTER-BASED OR ANY OTHER MECHANISM OR TECHNIQUE IS EXPRESSLY FORBIDDEN UNLESS SPECIFICALLY PERMITTED BY WRITTEN STATEMENTS WITHIN THIS DOCUMENT OR ANY OTHER DOCUMENT WRITTEN BY STRUCTURED SOFTWARE SYSTEMS LIMITED IN RELATION TO SPECIFIC PART OR PARTS OF THE SOFTWARE OR THIS DOCUMENT. ANY SUCH PERMISSION APPLIES SOLELY TO THE SPECIFIED PART OR PARTS OF THE SOFTWARE OR OF THIS DOCUMENT AND UNDER NO CIRCUMSTANCES DOES ANY SUCH PERMISSION APPLY TO ANY OTHER PART OR PARTS OR THE WHOLE OF THE SOFTWARE OR THIS OR ANY OTHER DOCUMENT.

## ACKNOWLEDGEMENTS

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organisations.

---

# About this Guide

## Description and Purpose

This is the Cradle Overview. Its aims are to:

- Explain what Cradle is and what it can do
- Show the Cradle architecture and each of the Cradle components
- Explain Cradle projects, their structure and possible contents

## Navigating the Guide Online

This guide contains hyperlinks when viewed as a PDF file. Selecting a hyperlink or the bookmarks on the left of the Acrobat Reader window takes you to the linked text. You can use these links and bookmarks to move quickly around the guide.

There are hyperlinks to:

- Each chapter from the list at the end of this section
- Each section and subsection from the Table of Contents

## Feedback

We welcome your suggestions for improvements to this guide. Please send them to:

**E-mail:**        [support@threesl.com](mailto:support@threesl.com)  
**Telephone:**   **+44 (0) 1229 838867**  
**Fax:**            **+44 (0) 1229 870096**  
**Post:**            **3SL**  
                      **Suite 2**  
                      **22a Duke Street**  
                      **Barrow in Furness**  
                      **Cumbria, LA14 1HH**  
                      **UK**

---

---

---

# Table of Contents

<b>About this Guide</b> .....	<b>i</b>
Description and Purpose .....	i
Navigating the Guide Online .....	i
Feedback .....	i
<b>Table of Contents</b> .....	<b>iii</b>
<b>Cradle-6</b> .....	<b>1</b>
What is Cradle? .....	1
What is in it? .....	1
What Can it Do? .....	2
Is it Extensible? .....	3
Is it Scalable? .....	4
What Can it Interface To? .....	4
Where Can it be Run? .....	6
Can it be Used by Remote Users? .....	6
How Can it be Deployed? .....	7
<b>Cradle Architecture</b> .....	<b>9</b>
Structure .....	9
Tools .....	10
Servers .....	11
Utilities .....	11
Interfaces .....	11
<b>Cradle Components</b> .....	<b>13</b>
Project Manager .....	13
WorkBench .....	14
Web Access .....	16
Cradle Database Server .....	18
Cradle Web Server .....	18
Plug-ins .....	19
Document Loader .....	21
Document Publisher .....	22

---

Spellchecker.....	23
<b>Scope.....</b>	<b>25</b>
Project Management.....	25
Information Capture.....	27
Requirements Management.....	27
Systems Modelling.....	28
Information Reporting.....	30
<b>Projects.....</b>	<b>33</b>
Cradle Projects.....	33
Project Databases.....	33
Items.....	34
Attributes.....	35
Categories.....	36
Frames.....	36
Cross References.....	37
Link Types.....	45
Link Groups.....	46
Pseudo Cross References.....	47
Link Rules.....	48
Ownerships.....	49
User Profiles.....	49

---

# Cradle-6

Summarises what Cradle-6 is, what it contains, what you can do with it, and where you can run it.

## What is Cradle?

Cradle-6 is the latest of the Cradle® Requirements Management and Systems / Software Engineering environments from 3SL.

Cradle is intended to be useful for projects concerned with:

- Requirements elicitation and definition
- System analysis and design
- System architecture definition
- Business process modelling and engineering
- Capability service definition and delivery

## What is in it?

Cradle fully supports Systems and Software Engineering processes with tools for:

1. Requirements Management
2. Systems Analysis
3. Architecture design
4. System and subsystem design
5. Function allocation
6. Performance assessment
7. Metrics
8. Test and acceptance management
9. Hazard, risk and critical issue management
10. Project document generation (paper, word processing software or web)
11. Configuration Management and control
12. Workflows

Cradle provides both web-based and non-web-based tools. All tools link to a central database server to ensure complete data consistency and integrity. With its distributed, multi-user architecture, Cradle has no need for any check-out, check-in or data merge operations.

Cradle is a multi-user environment in which groups of people work together in projects. The groups can be co-located, or geographically distant.

Cradle provides a comprehensive infrastructure to support all engineering processes:

- 
1. Extensible database, to support any project phase, include any type of data, and reference external data held in files, at URLs, or in other software environments
  2. Arbitrary scalability for project size, numbers of users, and distribution of users
  3. Built-in Configuration Management and control
  4. Optional security and access controls
  5. Optional user and team structures
  6. Integral electronic comments and reviews
  7. Internal or e-mail-based messaging
  8. User-definable project processes and phase hierarchies
  9. Workflows
  10. Web-based access from web browsers, and publishing into websites

## What Can it Do?

Cradle can be used as a full environment, or as one or more point tools. Although Cradle is a large product, you do not have to adopt all of it to use any part of it.

The major benefit of Cradle is when you need support for several project activities. Cradle integrates these activities in a single database. You do not have to move data between tools, nor rely on the existence of interfaces between tools, nor build such interfaces yourself where they don't exist.

Using Cradle you can:

1. Easily create a **Work Breakdown Structure** (WBS) to capture project activities
2. Create a **Cost Breakdown Structure** (CBS) to manage project and system costs
3. Quickly capture requirements using Document Loader and Word/Excel plug-ins
4. Engineer the requirements and generated URD/SRD/ORD documents
5. Build acceptance test cases and publish RTMs, CRVMs and PVMs
6. Build UML and functional models and allocate requirements into them
7. Evaluate alternative architectures and run performance assessments on them
8. Build test plans and test specifications and publish for distributed comments
9. Define **System Breakdown Structures** (SBSs) and **Product Breakdown Structures** (PBSs) and link to requirements, the system architecture and design
10. Allocate requirements and models to the architecture and simulate behaviour
11. Characterise system interfaces, and model and simulate protocol transfers
12. Generate SDS/SSDS documents and interface control documentation
13. Conduct software/hardware allocation and generate implementation specifications
14. Build software models, and code generate or reverse engineer software source code
15. Manage test definition and execution, integration test and acceptance
16. Collect customisable metrics at any stage in the project
17. Manage the process with metrics, CM, electronic/distributed reviews, alerts, annotations and email links

---

You can support all stakeholder groups with customised tool interfaces, queries, views, forms, and documents built with point-and-click tools. There are no complex scripting languages to learn.

You can publish requirements and models to websites or documents, or provide interactive view and/or edit access to them from users' web browsers. You can tailor these web environments to provide customised interfaces for each group of stakeholders.

You can build Product Breakdown Structures (PBSs) and System Breakdown Structures (SBSs) and link to corporate PDM / EDM systems, and create a Work Breakdown Structure (WBS), allocate all project activity within it, and a Cost Breakdown Structure (CBS) to manage system cost budgets.

## Is it Extensible?

Cradle is highly extensible and highly customisable.

All work that you do in Cradle occurs in a Cradle project. Each project has its own Project Database (PDB). You have full control over the structure of this database and the rules that operate within it. You also have full control over who can access the information in the database, which parts of it they can access, what they can do in these parts, and which tools they can use to operate on the database.

Cradle PDBs contain items, each of a user-defined item type, such as:

- User requirement
- System requirement
- Function
- Class
- Architecture component
- Hazard
- Issue
- Risk
- Business process

You can have up to  $10^{78}$  items of each of  $10^{78}$  types in each database. Items contain up to  $10^{78}$  attributes, each of which can store or reference up to 1 TByte of any kind of data. Attributes are called frames. The type of data stored in a frame is determined by its frame type. You can define up to  $10^{78}$  frame types.

Cradle can track all edits to each item in the database, and stores a full edit history for all attributes of all items.

Each frame either stores data, or contains a reference to data held externally from Cradle. You can link to external data held:

- In a file
- At a URL
- As an identity in an external environment such as a document management system

Each Cradle project has a set of login accounts called User Profiles for its users. Cradle allows these users to optionally be grouped into teams. Cradle provides a fully controllable access control system to manage who has what sort of access to which items in the database. This access control system includes multiple security classifications and privileges. Access control is also based on ownership, where every item in the

---

database is owned by either a user, a team or the project itself. Items owned by the project are held in project baselines, which are part of the built-in Configuration Management and change control system.

Each item supports an arbitrarily large set of comments, called annotations. A user does not need read-write access to an item to be allowed to add comments to it. Indeed, read-only users (users who have either chosen to log-in to Cradle read-only, or whose login has been forced to be read-only) can nonetheless add comments to items.

Items are linked by cross references. A cross reference between a pair of items indicates that they are dependent in some way. You can define any number of types of cross reference, called link types. You can link items simultaneously in any number of hierarchies, graphs or any other type of 1:1, 1:N or N:M relationships. Cross references also contain attributes, so you can store data inside each cross reference, such as:

- The reason for the link
- The justification for the link
- The current approval status of the link
- The coverage of the link

You can search the database only following links of a specific type. You can define any number of link groups, each a set of link types. Searching the database with a link group will search using all cross references whose link types appear in the link group.

You can define link rules to control who can operate (create, edit, delete) links of specific types between items of specific types.

The structure of a project's database, defined by the item types, link types, frame types and so on, and the access rights, user accounts and project rules is collectively called the project schema. You can define a standard project schema that accurately reflects your development process, and use this schema in all new or existing projects. You can have a specific schema for each project.

All of this extensibility is through point-and-click UIs. You do not have to learn any kind of complex scripting language to be able to extend Cradle.

## Is it Scalable?

Cradle is highly scalable.

You can create any number of Cradle projects, all of which can be accessed simultaneously. Each Project Database can be of any size, with a theoretical upper limit of  $10^{235}$  TBytes.

You can have up to 8,192 concurrent users of a Cradle system.

Despite this massive scalability, Cradle is perfectly suitable for small teams, or even single user projects.

## What Can it Interface To?

You can link Cradle to other tools and embed Cradle within other tools. Cradle integrates with your existing desktop applications, and larger environments such as CAD, document management and EDM/PDM.

Cradle has a variety of interfacing mechanisms, including:

1. Ability to link to any application and encapsulate or link to their data. 3SL supplies pre-built interfaces to products such as:
  - a. Microsoft Word®
  - b. Microsoft Excel®
  - c. Microsoft PowerPoint®
  - d. Adobe Acrobat®
  - e. Adobe FrameMaker®
  - f. FileMaker®
  - g. Adobe PhotoShop®
  - h. Adobe Illustrator®and you can easily add your own interfaces through a point-and-click UI
2. DDE interfacing (Windows only)
3. Support for a variety of import/export formats, including:
  - a. **Cradle** (ASCII-based and easily parsed and used in other applications)
  - b. **CSV** (comma separated value)
  - c. **AP233** (part of ISO 10303)
  - d. **XMI** (for UML model exchange)
  - e. **XML** (Cradle XML format)
  - f. Other third party formats
4. Data Converters for other applications, including:
  - a. Teamwork®
  - b. Select®
  - c. CORE®
  - d. BPWin®
  - e. RDD-100®
5. An Application Programming Interface (API) for C/C++ and VB/VBA (Windows only)
6. Open, and published database formats, with databases built on industry-standard ISAM files, accessible through a variety of external applications including IBM DataBlade®
7. A variety of 3SL-supplied utilities to manipulate external data in user-defined ways so that it can be brought into Cradle, or manipulated once Cradle has produced it
8. Plug-ins for Microsoft Word and Excel (Windows only) through which paragraphs, tables, figures, rows and columns can be imported into Cradle
9. A plug-in for TeleLogic's DOORS® product to import information and cross references from DOORS into Cradle and from Cradle into DOORS (the plug-in is fully bidirectional)
10. Publishing to Word, Excel and FrameMaker
11. Website publishing to standard (non-Java) HTML with SVG (Scalable Vector Graphics) used to publish diagrams

Cradle supports all current forms of Microsoft Office®:

- 
- Office XP
  - Office 2003
  - Office 2007

## Where Can it be Run?

Cradle can be run in any configuration of Windows® or UNIX® environments:

- PCs and servers running Windows 2000, XP, Vista, Server 2003 and Server 2008
- Workstations and servers running UNIX:
  - HP-UX® 11 for PA-Risc processors
  - AIX® version 5.1 or later
  - Linux (Kernel version 2.4 or later), such as Debian, Fedora, Tau, RedHat Enterprise 3
  - Solaris® version 2.8 or later

Any part of Cradle can run on any supported operating system. You can mix operating systems if you need to. You can move data from UNIX to Windows and vice versa without restriction.

## Can it be Used by Remote Users?

Cradle can be used by remote users, whether they are accessing the main Cradle system from a remote location, or accessing a local copy of the central Cradle Database.

There are several options for linking into the main Cradle system when you are remote from it:

1. Running Cradle components locally with a connection to the remote Cradle Database Server (CDS) over:
  - a. The corporate LAN or WAN
  - b. VPN connection directly into the main network
2. Running Cradle components on the main system but displaying locally to the user (ASP)
3. Web browser connection running through a VPN link or through a secure `https` connection to the Cradle Web Server (CWS)

Remote users can also work on a local copy of some or all of the main database. Cradle can ensure that any work that you do when remote from the main system is not in conflict with work being done on the main system. There are several options, including:

1. Forcing read-only access to parts of the database
2. Forcing unique IDs for work being done remotely
3. Marking sections of the main database as checked-out

These same facilities can also be used when separate databases are being worked on by multiple groups in different locations, such as partners or subcontractors, or the project customer wanting visibility of the work being done.

Naturally, 3SL recommends that only a single instance of a Project Database is used, with remote groups connecting into this central database using any of the options listed above. However, we fully recognise that

---

there are situations in which such electronic links are not practicable, particularly where the work is for the military, and the content of the Cradle Database is classified.

In situations where multiple instances of a Project Database (PDB) are unavoidable, the above techniques can be used to prevent concurrent conflicting updates by the separate groups. In addition, Cradle has a number of mechanisms to ensure that the multiple instances can be kept synchronised, including:

- Automated import/exports
- Selective export and import of comments on items, rather than the items themselves
- Reports of activity in the database
- Automatic messaging system

## How Can it be Deployed?

Cradle supports all possible application deployment strategies, including:

1. Local installation on end user's desktop machines
2. Run on desktop machines but retrieved from a central network installation on a file server
3. Run on a remote server and delivered to the desktop using redirection:
  - a. X Window System®
  - b. Windows Terminal Server®
  - c. Citrix WinFrame® and MetaFrame®
4. Run remotely through a VPN or PPP dial-up connection

Cradle fully supports all network topologies and the use of firewalls and DMZs. Cradle uses only basic TCP/IP protocols (UDP and TCP), and has its own internal communication protocol including optional message compression and encryption. All Cradle communications use controllable ports that can be set to comply with any firewall management requirements.

Cradle does not depend on any proprietary network services (such as NIS, RPC, NFS, LDAP, Active Directory or ASP). Cradle does not use Java, nor applets or tclets. Cradle will use any naming service that it finds to be running, for example for name resolution and IP address translation.

Cradle software components and PDBs can be housed on any filesystem, including NAS (network attached storage) devices, Windows shared drives, Novell filesystems and UNIX NFS mounts.

Cradle can be remotely deployed and supports silent installs. It is fully compatible with Microsoft SMS®.

Cradle's servers are designed to be **start-and-forget**. They can run 24x7 with no need for periodic shutdown for backups, or any other purpose. Cradle provides command-line backup utilities that can be used to complement main image backups. These utilities can be run in automated batch jobs.

---

---

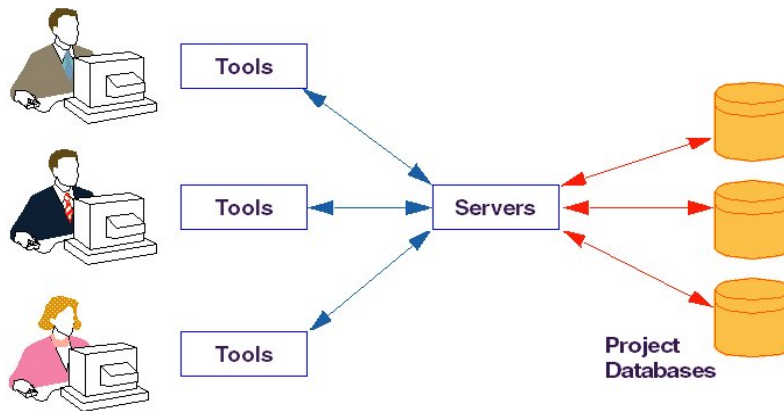
# Cradle Architecture

## Structure

Cradle is a multi-user Requirements Management and Systems / Software Engineering environment with which users can work cooperatively to create, update, manage, interlink and report any and all types of information involved in modern complex projects.

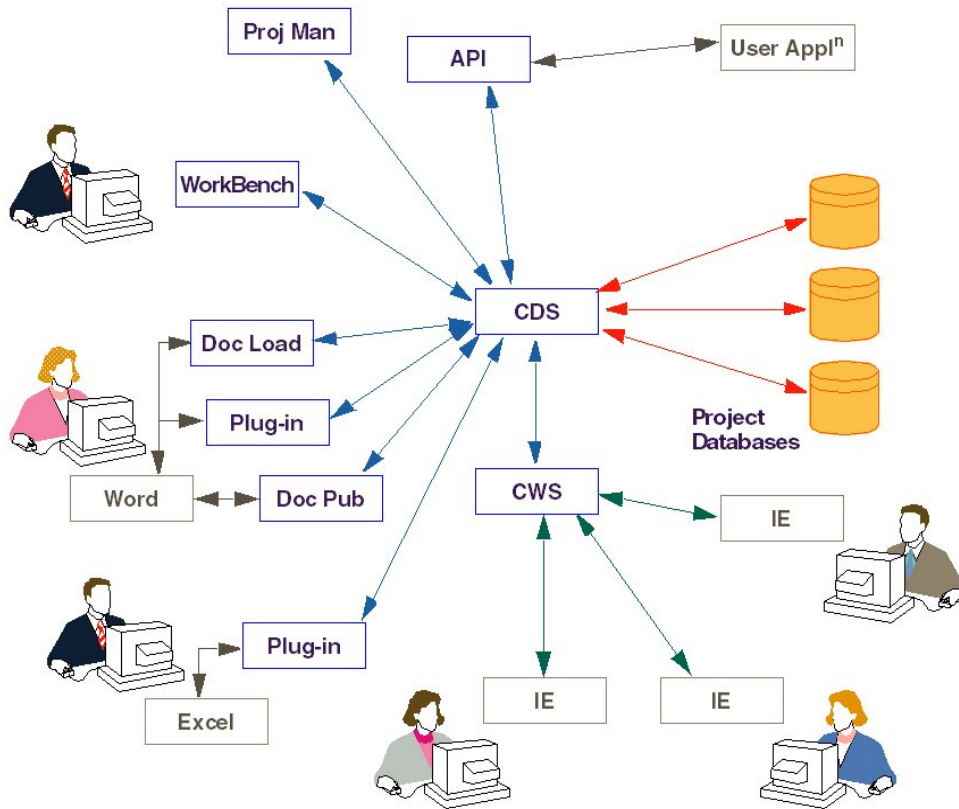
Cradle is a client-server product, meaning that it has two major types of components:

- Front-end tools
- Back-end servers



The tools are what people use, and consist of the Cradle tools and web browsers. The servers read and write data to and from Cradle Project Databases (PDBs) and create a multi-user environment. Each tool connects to a server. No tool ever accesses a database directly, they always use a server.

There are also some utilities and interfaces that can be used to connect Cradle to other applications and environments. The Cradle architecture is:



## Tools

There are many tools in the Cradle environment, for Requirements Management, Systems Modelling, Risk Management, Document Generation, Reverse Engineering, and so on. These tools are grouped into three Cradle components:

- Cradle WorkBench
- Project Manager
- Microsoft Office® tools, including a Spellchecker, Document Publisher, Document Loader and plug-ins for Microsoft Word® and Excel®

Cradle also has a Web Access capability that allows users to access Cradle Databases over a corporate intranet or the Internet. Web browsers such as Microsoft Internet Explorer® can also be considered to be Cradle tools.

All work that you do with a Cradle tool is in the context of a project. The default structure (called the schema) for all new Cradle projects is described in a separate Default Project Schema document in the Cradle documentation set.

---

## Servers

There are two Cradle servers:

- Cradle Database Server (CDS)
- Cradle Web Server (CWS)

The CDS is the most important Cradle server. It reads and writes data to and from Cradle Databases. All Cradle systems have a CDS running somewhere. The CDS provides all the facilities that are provided in standard database management systems, but with the advantage that no database administration is needed to operate, maintain or manage the CDS. Simply start it and then forget about it.

The CWS links users' web browsers to the CDS. In effect the CWS:

- Receives user instructions from their web browsers
- Passes them to the CDS to manipulate the Cradle Database(s)
- Reformats the results into HTML pages that are sent to users' web browsers

The CWS delivers web-based UIs to people who want to access Cradle PDBs from their web browser. These users can be local or remote.

The CWS can present different web UIs to different groups of people, if you want it to. For example, you could define different types of users and present users of each type with a web UI that is tailored to their needs and interests.

## Utilities

Cradle is provided with a range of command-line and other utilities to make it easy to administer a Cradle installation and easy to manage Cradle in your overall IT environment:

- Project Manager for project administration
- Command-line tools for automated backups
- Diagnostics to identify and solve problems

## Interfaces

Cradle is a large and very capable product and does not have to be used in isolation in any project. Cradle can be interfaced either to other tools that Cradle can launch (such as Word and Excel), or other environments of which Cradle is a part.

Interfaces exist to:

- Connect Cradle to common desktop tools
- Exchange data with other tools that you use as part of an overall SE solution
- Allow you to create customised tools and utilities of your own through an API

---

---

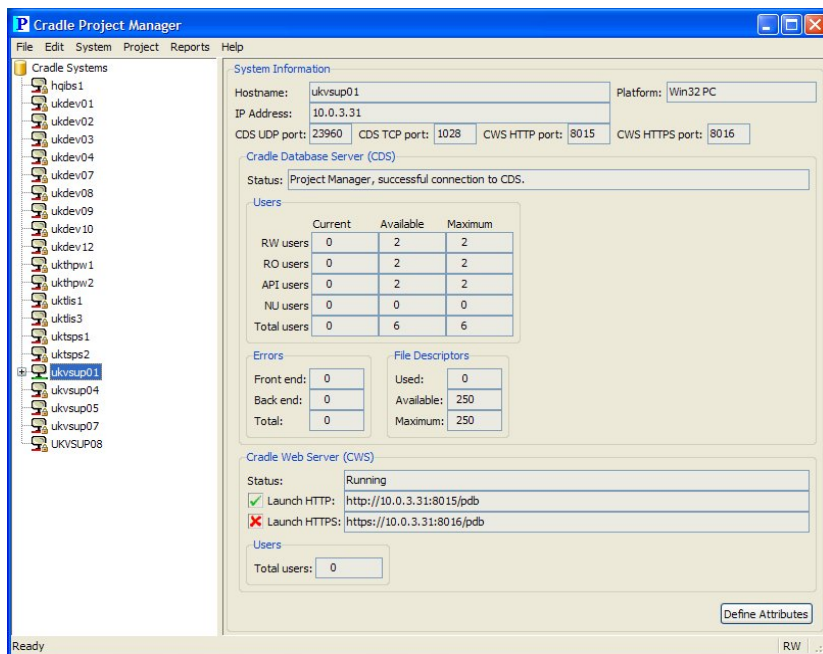
# Cradle Components

The Cradle components are explained below.

## Project Manager

Project Manager is a utility which can be used to manage your Cradle projects and both the Cradle Database Server (CDS) and Cradle Web Server (CWS).

It allows a project hierarchy to be defined that contains some or all of the projects defined in the project register of a single Cradle system, grouped into any tree structure that is meaningful to the users of that Cradle system.



Project Manager finds all CDSs visible to it, either by broadcasting onto the local TCP/IP subnet, or by referring to a single machine whose TCP/IP hostname or IP address has been set in the **CRADLE\_CDS\_HOST** environment variable or on the command line [`-cds n.n.n.n | hostname`].

Attributes can be set to characterise a project. These are defined in the project hierarchy for all projects in a Cradle system, and also for each project. The project hierarchy defines:

1. What attributes are to be held for each project
2. The values of these attributes for each individual project listed in the project hierarchy

---

### 3. The hierarchy of these projects

A default set of project attributes is provided in the project hierarchy. This default set is modifiable by appropriately authorised users that are defined in the `system_users` file.

For further information on Project Manager please see the Project Manager User Guide.

## WorkBench

WorkBench is Cradle's primary Systems Engineering tool interface that allows you to customise exactly what you need to view and manipulate in the database in the format you wish to output it.

WorkBench contains all modules in the Cradle system except for WEBA (Cradle Web Access). It is used by Systems and Software Engineers, by Team Leaders, Configuration Managers and Project Managers.

Project tools can:

- Generate textual reports or user-defined documents
- Build arbitrarily many implementation-independent analysis models and arbitrarily many architecture models and implementation-dependent design models
- Check static or dynamic consistency of analysis, design or architecture model data
- Generate websites from project data
- Gather project metrics
- Convert detailed designs into program code designs and code generate such detailed designs

Project administration is available, providing the means to define the project's conventions and database structure (collectively its schema) and the User Profiles of project members. This also includes the built-in Configuration Management System (CMS) through which Cradle's QA/QC mechanisms can be used for:

- Formal review and approval
- Baseline and version control
- Formal change control
- Audit trails

When used as a Requirements Management tool, WorkBench provides all Requirements Management tool support needed by Project Engineers. The only other capability that may be needed is the ability to programmatically parse external source documents, capture and link them to items in the database, and automatically identify differences between such external documents.

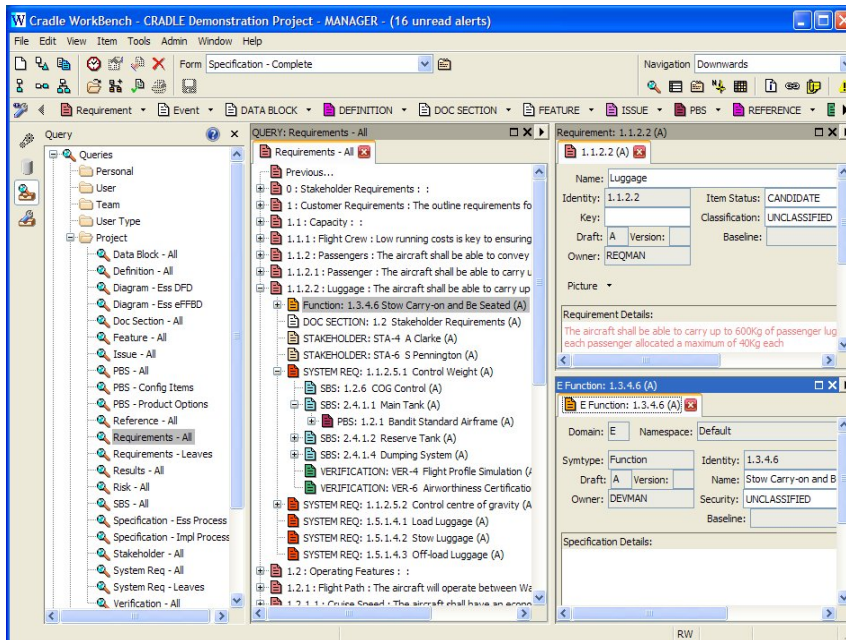
---

**Note:** This method for programmatically capturing data from external documents and creating cross references between paragraphs in external documents and items in the database, is done using Cradle's plug-ins for Word and Excel and Document Loader.

---

The **File** menu provides import/export facilities for project data in many formats.

WorkBench provides extensive customisation, hierarchy and table display, drag and drop, and query facilities, particularly for creating cross references between items.



The left panel in the WorkBench UI contains four sidebars which are:

- **Project** - the entire contents of the database
- **Phase** - phase hierarchy
- **Query** - all user-specified, project and system-wide definitions
- **Control** - controls for drawing diagrams and Hierarchy Diagrams (HIDs)

Items can be dragged from this display and dropped into the right part of the UI, where they are opened in user-defined forms. Double clicking a query item from the **Query** sidebar will open a new pane, displaying the query results in a user-definable view.

All the WorkBench UI can be customised. Sessions can be defined that pre-load the UI with queries and associated views, and specific sizes of the WorkBench window and the panes within it. Sessions can be selected when users log into their project in WorkBench.

WorkBench provides list, table, document and hierarchy displays of project data. Tables present a set of items found in the database from a user-defined query, one row for each item. User-defined views control which attributes are shown in the columns. Tables can show cross referenced items as nested rows, and can have any amount of such nesting. Any number of tables can be shown and manipulated at the same time. Tables can be published to RTF or CSV (for Word) and HTML (for web browsers and Excel), using predefined table styles that control colours and fonts.

All table cells are editable (subject to access rights), so WorkBench can have a familiar, spreadsheet style appearance. Items can also be displayed in user-defined forms, to present some or all of their attributes in a customisable form-fill display for use by non-engineering project personnel.

---

In hierarchies, WorkBench presents a set of items in Explorer style tree views. Double clicking an item shows the items that are linked to it, as for files in a folder. This process can be repeated as many times as desired. Drag and drops between trees creates cross references.

WorkBench can be started from the plug-ins, Project Manager or launched directly.

For further information on WorkBench please see the WorkBench User Guide.

## Web Access

Web Access allows users to gain access to Cradle Databases by directing their web browsers to a Cradle Web Server (CWS). This allows them to gain access from anywhere on the LAN, WAN or across the Internet. Such users do not need any Cradle software installed on their machine, Cradle does not require the installation of a Cradle-specific web browser plug-in.

The tool provided to web users is really the contents of web pages displayed in their browsers. These pages are created by the CWS from:

- A set of HTML templates, one for each type of item in the database
- A set of page designs supplied with the CWS
- The items in your database

The CWS dynamically creates a website containing data drawn from the user's Project Database (PDB). The website provides facilities that allow web users to interact with this database information, to query it, view or edit it, and navigate from it. Exactly what facilities are provided to the user are determined solely by the contents of the HTML templates used to define the form and content of this website.

You have full control of the contents of these HTML templates, so that you can control:

- What the web interface looks like
- What facilities it provides
- How these facilities are accessed

You can create different sets of HTML templates for different web interfaces to the same project. A set of templates is associated with a specific user class. You can define any number of user classes, and then create a set of HTML templates for each. You assign each Cradle project member to a specific user class, which defines which of your web interfaces they will see.

This capability can be used to provide customised web interfaces to different stakeholder groups, such as customers, partners and subcontractors.

The website generated by the as-supplied HTML templates provides query and browse facilities:

ID	Key	Ver	Dft	Name	Type	Priority
0			A	Stakeholder Requirements	ACCEPTED	
1			A	Customer Requirements	AGREED	HIGH
1.1			A	Capacity	AGREED	HIGH
1.1.1			A	Flight Crew	CANDIDATE	HIGH
1.1.2			A	Passengers	AGREED	HIGH
1.1.2.1			A	Passenger	CANDIDATE	HIGH
1.1.2.2			A	Luggage	CANDIDATE	HIGH
1.2			A	Operating Features	AGREED	MEDIUM
1.2.1			A	Flight Path	AGREED	MEDIUM
1.2.1.1			A	Cruise Speed	CANDIDATE	MEDIUM
1.2.1.2			A	Runway Length	CANDIDATE	HIGH
1.2.2			A	Flight Schedule	CANDIDATE	MEDIUM
1.3			A	Environment	AGREED	
1.3.1			A	Wind	CANDIDATE	MEDIUM
1.3.2			A	Temperature	CANDIDATE	MEDIUM
1.3.3			A	Humidity	CANDIDATE	LOW
1.3.4			A	Precipitation	CANDIDATE	HIGH
1.3.5			A	Ground icing	CANDIDATE	MEDIUM
1.4			A	Dependability	AGREED	HIGH
1.4.1			A	Maintenance	CANDIDATE	MEDIUM
1.4.2			A	Repair	CANDIDATE	HIGH
1.5			A	Regulations	AGREED	MANDATORY
1.5.1			A	CAA Regulations	AGREED	MANDATORY
1.5.1.1			A	Airworthiness Regulations	ACCEPTED	MANDATORY

The Web Access module provides a Master Tree presenting a hierarchical view of all items of all types in the database. Double clicking in this tree follows cross references to or from an item to the items linked to them. Selecting an item opens it in a window for viewing or editing, depending on the user's access to the item.

You can define, create, save and run named queries. A set of default queries is provided. You can create any number of new queries.

Running a query retrieves from the database all items that match the query's criteria, and presents the items as rows in a table. The columns in the table are determined by a Web Access view. You can create any number of views and select them from the **Views** combo box.

Depending on which view is selected, each row may provide controls to view or edit its item, to delete it, or to follow cross references to all items that are linked to the item and display these linked items in a further table. There are also controls available whereby you can create children, siblings and cross references. Navigating through cross references can be continued to any depth.

Items are viewed or edited in user-defined forms. To prevent simultaneous updates by other WorkBench or web users, items are locked during such edits.

The default website provides administrative facilities for authorised users from the  button.

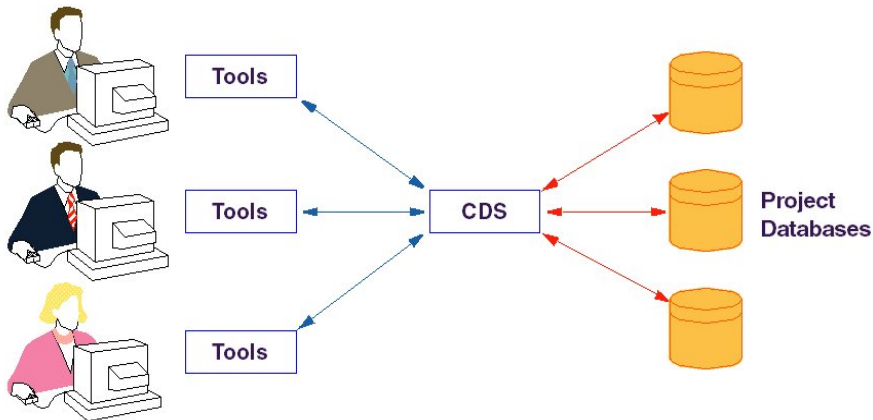
Web Access can also be launched from Project Manager.

For further information on Web Access please see the Web Access User Guide.

---

## Cradle Database Server

The Cradle Database Server (CDS) allows tools to interact with Project Databases:



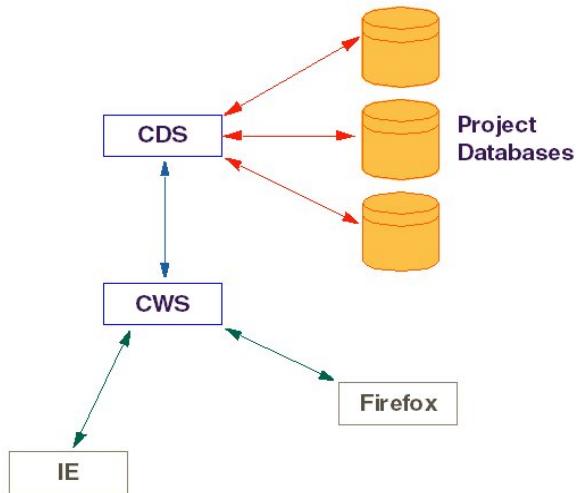
Users can dial-in to the CDS from laptops linked to a telephone line, subject only to the security restrictions of your network. If an electronic link can be created from a tool to the CDS, data can be managed by tools anywhere. It is the ability of the CDS to support concurrent users (up to 8,192) that makes Cradle a multi-user system.

These options for tool access are central to Cradle's use as the infrastructure for Virtual Integrated Project Teams, when groups collaborate on a project, but the groups are geographically distributed, and are typically part of separate organisations (such as the purchaser, the user, the prime contractor and the sub-contractors).

The CDS is a server because it runs in the background. This means that unlike most programs, there is no UI with buttons and menus that you can see. On UNIX machines, this type of program is called a daemon. On Windows 2000, XP, Vista, Server 2003 and Server 2008 this type of program is called a service.

## Cradle Web Server

The CWS links users' web browsers (such as Microsoft Internet Explorer) to the Cradle CDS and then to the data in Cradle Project Databases.



The CWS contains a small web server, together with Cradle-specific code. It:

- Receives instructions from web browsers
- Passes them to the CDS to manipulate the Cradle Database(s)
- Re-formats the results into HTML pages which are sent to users' web browsers

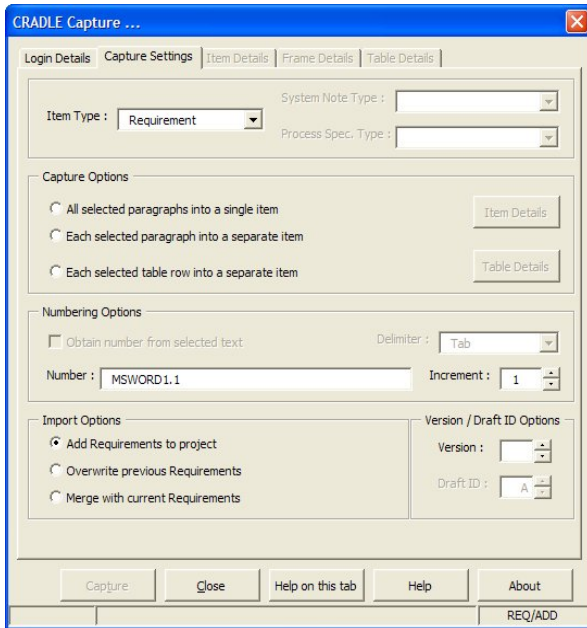
The CWS effectively creates a website from its user-definable HTML templates and layouts, and the items in the database. This website contains facilities to manipulate items and links between items, and to define and run queries on the database. Users' web servers render this website to make Web Access available to web users.

There is only one CWS in a Cradle system. It manages all web users and a set of connections to the CDS that interacts with PDBs on behalf of the CWS, and hence the web users.

As the CWS is a server, it runs in the background. It has no UI, but does provide an administration capability in the website sent to users' web browsers that allows appropriately authorised users to administer the CWS itself. On UNIX machines, the CWS runs as a daemon. On Windows 2000, XP, Vista, Server 2003 and Server 2008 this type of program is called a service.

## Plug-ins

Cradle provides plug-in modules for Microsoft Word and Excel. These are loaded when Cradle is installed on PCs that have Word or Excel 2003, XP or 2007 already installed on them:



The plug-ins provide facilities to capture data from documents and spreadsheets into items in the database. You can capture:

- All selected paragraphs into a single item
- Each selected paragraph into a separate item
- Each table row into a separate item
- Each spreadsheet row into a separate item

Each plug-in has a **Capture...** dialog through which you log-in to your PDB. This loads the project schema into the plug-in so you can capture into any type of item defined by the project, and select attribute values from combo boxes. Auto-increment facilities are provided to capture sets of items at once. You can keep the plug-ins' windows active whilst manipulating Word and Excel, so you can perform a series of captures without repeatedly logging-in to a project.

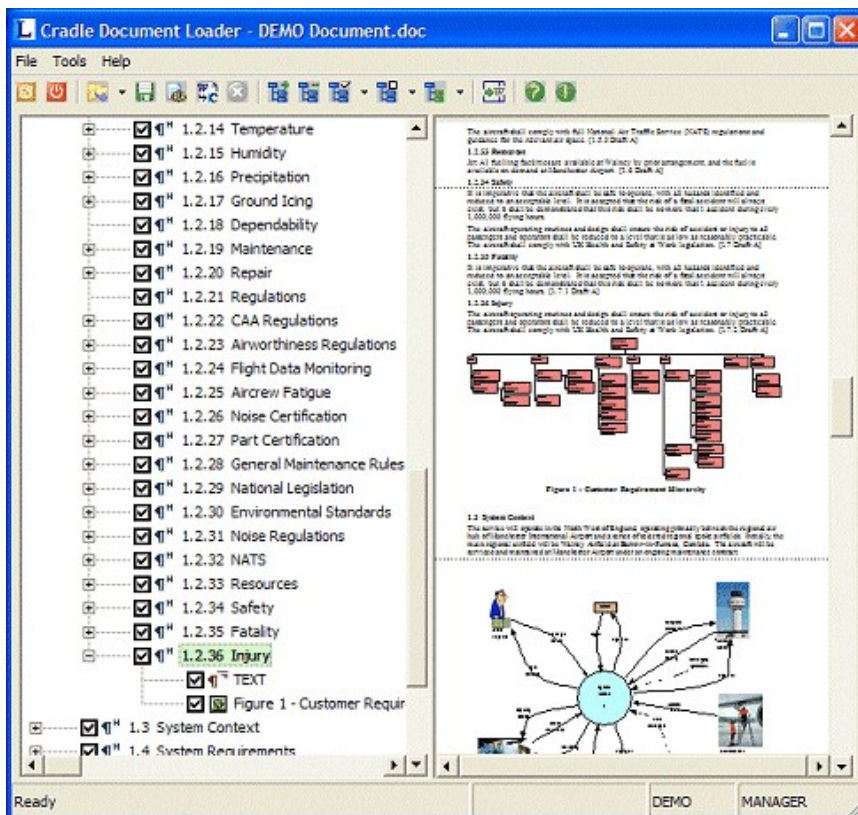
When capturing from tables or spreadsheets, you define a mapping between columns and attributes of the database items that specify where each column's data should go. You can also ignore columns.

You can use the Word and Excel plug-ins cyclically, to either store new information in the database, or to merge information from a Word document into existing data. This allows you to, for example:

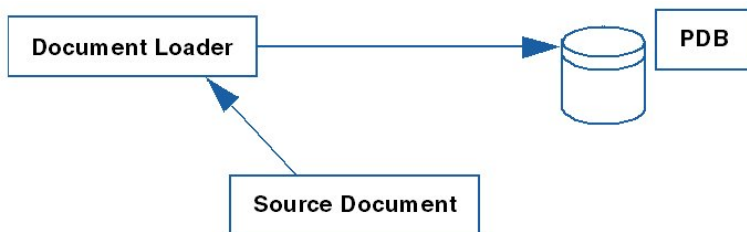
1. Capture information from a Word document
2. Modify the information in Cradle
3. Export the information as a Word document or table
4. Modify the information in Word
5. Merge the changes back into the Cradle Database

## Document Loader

Document Loader is shipped as part of the Office Toolsuite. The basic concept of Document Loader is taking a source document and capturing its contents based on its paragraph numbering as a hierarchy of items and frames which can be imported into a Cradle PDB.



**Note:** Document Loader is only compatible with Word, i.e. it cannot be used with PDF files.



---

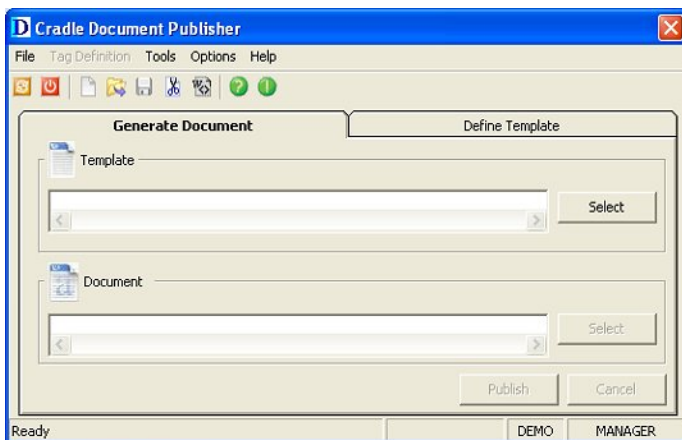
Cradle stores the source document in the database allowing you to trace captured items back to the source. New versions of these documents can also be controlled within Cradle. Document Loader can capture body text and figures as well as associated frames, and capture individual table rows or cells as items. Information can be captured into requirements, specifications or system notes. Microsoft Word's paragraph formatting is used in order to create the hierarchy of headings and subheadings. The source document will typically contain requirements, system notes or specifications grouped as a series of numbered paragraphs. Document Loader then parses the document, extracts the numbered headings and uses these to create the items. Related subheadings will have item cross references created automatically. Normal text, pictures or tables within the paragraph will be imported as associated frames.

Document Loader can be started from Project Manager or launched directly.

For further information on Document Loader please see the Document Loader User Guide.

## Document Publisher

On Windows-based Cradle systems, Document Publisher is used to generate any documents from Cradle through Microsoft Word. The form and content of these documents is completely controlled by a document template created in Microsoft Word. All the formatting and style aspects of the final documents are controlled through page setups, paragraph and character styles created in Word. In this way, Document Publisher can use your existing Word templates, ensuring that the documents it produces are fully consistent with the rest of your documentation.



The information content of the documents generated by Document Publisher is controlled by tags that you insert into the document template. The contents of these tags are instructions that Document Publisher uses to extract items from the PDB and then to select the attributes of these items that are to be printed, and to control how they are to be printed (such as paragraph styles and numbering to be used).

By using the facilities built-in to Word, the documents produced by Document Publisher can include Tables of Contents, Lists of Tables, Figures, Tasks and any other structuring elements that you require.

All aspects of the output documents are specified through the point-and-click UI provided by Document Publisher. There is no need to learn any form of complex scripting language.

Cradle Document Publisher is part of the Cradle Office Toolsuite. There are different versions of this toolsuite for each of the versions of Microsoft Office supported by Cradle, which are:

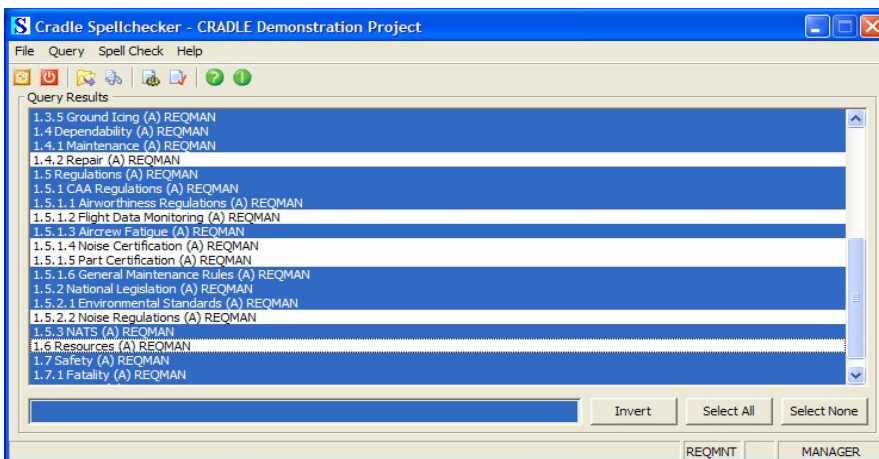
- Office XP
- Office 2003
- Office 2007

Document Publisher can be started from Project Manager or launched directory.

For further information on Document Publisher please see the Document Publisher User Guide.

## Spellchecker

Cradle provides a Spellchecker utility that can be launched as a standalone application, or can be started from WorkBench or Project Manager:



However started, Spellchecker checks the spelling of a collection of items. The items to be checked are those found by running a query, either the current query (when it is started from WorkBench) or a query chosen by the user (when started standalone). These items are shown in the Spellchecker UI, where you can select or de-select items to further refine which items are checked.

You can control which attributes, frames and categories for the items found by the query are to be spell checked.

Spell checking uses the same dictionaries and dictionary options as Microsoft Word.

Any errors found which you choose to fix whilst spell checking are immediately written back into the database.

---

---

---

# Scope

Describes the scope of Cradle-6.

## Project Management

In any project, once information is created it undergoes a process of continual change as evolving customers need impact on the work completed thus far, and as the work of later project phases necessitates changes to the deliverable of earlier phases.

Changes must be minimised to maximise productivity. Once active, all changes must be managed so they have minimal impact on work in progress. The ability must exist to regress to an earlier known state if recent work proves to be nugatory. Similarly, a record must exist of the evolution of all project items and the process by which each evolved to its current state.

In larger projects, the flow of deliverables between project teams, and from partners and sub-contractors to a prime contractor must be controlled. Flexibility must exist to respond to changing circumstances.

The Cradle Configuration Management System (CMS) is a flexible, customisable framework from which a project can create a specialist Configuration Management (CM) environment. The CMS provides four basic mechanisms:

- Formal review
- Baselines and version control
- Formal change
- Audit

Collectively, these mechanisms can control the development of information and manage changes to it. They are both necessary and sufficient to provide AQAP, BS or ISO compliant change management processes, and also to meet the needs of a Change Control Board (CCB).

The CMS operates on individual items of information. External CM systems (PCMS - Product Configuration Management System, PVCS - Polytran Version Control System, SCCS - Source Code Control System, RCS - Revision Control System, CVS - Concurrent Version System and so on) work with files, and cannot offer the same degree of control as the Cradle CMS. Moreover, as items in the Cradle Database can reference external files, the CMS can complement such file orientated CM tools.

The operation of the CMS is controlled by privileges that can be granted or denied to project members to reflect their roles in the project. These can be changed at any time. External IV&V is supported in the review and approval process.

The CMS can be tailored to the project. As Change Requests and Change Tasks are themselves items in the database, they can contain attributes to store existing change control documents. The enforcement of project rules and checks can be fully automatic.

Any development process can be supported by combining the CMS and the Cradle workflow facilities, for example:

- 
- The CMS is to be a record keeper of the results of review meetings
  - Reviews are fully electronic, no meetings
  - Reviewed information enters a baseline
  - Reviewed information moves through project teams in a defined sequence before final approval into a baseline
  - Reviewed information is passed to sub-contractors for further development and refinement, with notifications of change being managed by the prime contractor
  - Separate development tools pool related information within and between themselves prior to publishing the material for universal use within the project

Records of all reviews are automatically maintained, including review comments, whose inclusion can be made mandatory. Once baselined, information is protected against modification, and can only be updated through a formal change.

If a project uses Cradle's team mechanism, then the first review of new or changed items occurs within the team. A successful review elevates the items to being owned by that team, making them available read-only to all team members, but not normally accessible outside of the team. This allows each team to have its own team baseline, of reviewed material. Items can be reworked and amended within the team baseline as often as needed. When appropriate, some or all of a team's baseline can be reviewed at project level for entry into a project baseline.

Special user privileges allow Change Requests to be raised and approved, and to action Change Tasks with reference to approval Change Requests. Items are changed under one Change Task at a time, but can be moved between Change Tasks. Management of changes is normally a CM responsibility. Proposal of changes is usually limited to senior engineers and customer interfacing staff. Sanctioning of changes, and the management of the whole change process is normally confined to the CCB. As each of these functions requires separate privileges implementing such organisational schemes in the Cradle CMS is very straightforward.

The contents of the latest project baseline are available, read-only to all users with the necessary privileges. These items remain accessible until the next baseline is closed. Unchanged items will migrate from the old baseline into the new. Closing a new baseline does not create copies of unchanged items that migrate from the previous baseline, nor are deltas created. This ensures that the baseline mechanism is fast, and economical in its effect on database size. Creating a baseline also makes a copy of all cross references between the baseline items, so that any subsequent changes to the cross references between items do not impact on the integrity of the baseline.

Information cannot be deleted from a baseline, but can be retired. Retirement of an item means that it will not migrate into subsequent baselines, and thereby effectively becomes deleted from them. Retired items can be reinstated.

Restoring a baseline restores the versions of items that were originally in it, and all cross references that existed at the time that the baseline was originally created.

Completing a Change Task automatically completes all Change Requests that are associated with it for which this is the last incomplete Change Task.

The change status of a baseline can be reported in terms of the Change Requests and Change Tasks that have been completed within it. The delta between baselines can be reported, in terms of new, updated and retired (deleted) items.

---

A wide variety of CM report types, styles and options allows all aspects of the project, and its history, to be reported by any user. These include the means to report current item reviews, the status of all current changes, and the entire life history of any specific item, any class of item, or all types of item.

## Information Capture

Cradle has a comprehensive range of Requirements Capture tools to assist or automate the capture of user and operational requirements from external documents (text, Word, Excel etc.), other tools, and raw data files. You can also enter requirements into Cradle directly.

You can register external documents into Cradle and customise how requirements are captured from them. You can capture several times if needed, each time either replacing some of the requirements that you have already captured, or merging new data into your existing requirements. Capturing from registered external documents creates links (cross references) between the document and the requirements captured from it.

You can register new versions of these external documents. Cradle finds the differences between the old and new documents, and, where appropriate, highlights requirements linked to the changes so that changes can be made.

You can use these techniques to load other types of information as well, such as domain knowledge, assumptions and statements from industry standards.

## Requirements Management

Cradle regards Requirements Management as the combination of requirements capture (also requirements elicitation) and Requirements Engineering, over time, in the context of a Configuration Management and change control system.

You can create multiple sets of requirements for different purposes and link them:

1. **Business requirements** that express the business goals for the system and put the system into context with other systems and developments
2. **User requirements** that express what the customer expects the system to do
3. **Operating requirements** that express how the customer expects the system to work
4. **System requirements** that express constraints on what the system will contain, or how it will be implemented
5. **Derived requirements** that express system requirements in the context of a specific physical equipment for a selected architecture and implementation

You can sub-divide, or sub-group, these sets into requirements of different types:

- **Functional**
- **Performance**
- **Environmental**
- **Human factors**, and so on

and into different levels, such as:

- **Force**
- **Platform**

- 
- **System**
  - **Zone**
  - **Compartment**
  - **Subsystem**
  - **Equipment**
  - **Assembly**
  - **Component**
  - **Part**

You can have any number of these organisational or descriptive attributes in requirements, and indeed in any other type of information in the Cradle Database.

You can engineer the requirements to remove ambiguity, imprecision, contradiction, duplication and omission. Requirements can be derived, grouped and sub-grouped, split and merged. You can find missing requirements from coverage analyses and by linking requirements into later project phases, such as analysis and architecture models.

You can create satisfaction arguments that explain how one set of requirements (such as system requirements) have been created from an earlier set (such as user requirements). These satisfaction arguments can be in different forms, including:

- Attributes in the cross references between the requirements in each set
- Analysis models
- Separate items in the database that link one or more requirements from one set to one or more requirements from the other set

## Systems Modelling

Cradle has a full suite of graphical modelling tools with which you can build business process, analysis, architecture and design models, as well as representing the SBS and PBS. Cradle has separate analysis and design domains so you can build models for different purposes and link them.

Each modelling domain contains a user-defined model hierarchy which is a hierarchy of models. Each model hierarchy consists of named model levels.

Different types of models are needed for different processes. Some processes require a suite of models to be built, each model representing the system and its environment from different perspectives.

Whilst requirements are statements of need, models are expressions of how these needs are satisfied. Analysis models are often regarded as graphical requirements. Design models are representations of the system. Both express the final system in greater detail and with greater realism. The final set of models are the engineering drawings and code structures from which the system is manufactured.

A model can also be a representation of requirements when used as a requirement engineering technique rather than a system design technique (a hierarchically structured system requirement can be represented by a functional flow model). A graphical representation at the abstract level eases stakeholder review and allows exploration of various concepts and ideas. Explicit linking of functional requirements exposes the interactions between the functions (the start of interface specifications) and highlights missing or excessive functional requirements. Modelling the system requirements can demonstrate that the requirements are complete and consistent.

---

Models are indispensable and no system should be developed without them. The number and types of model needed for a system depends on its complexity and form.

You can create all of the models needed by your system in Cradle, such as:

- Environment models showing the interaction across the system boundary
- Mission profiles or usage scenarios for time-sequenced views of system behaviour
- The systems of systems viewpoint showing how the system under study interacts with other systems in its environment
- Process models to show how the system under study fits into the organisation, its people, and existing procedures
- Analysis models to depict system functionality in an implementation-independent manner
- Architecture models to show system structure, subsystems, equipments and components, and the allocation of functional or executable elements to each
- Interface models to show the data exchange protocols
- Design models to show functionality, data, or classes (UML) of a system element
- Operational models to show the human-to-system operational sequences
- Implementation models to show hardware or software designs, fabrication or construction drawings, or runtime execution schematics

Models can be created using any combination of many different notations, including:

- Functional (such as IDEF0 - Integrated Definition for Functional Modelling, Data Flow Diagrams - DFDs, and State Transition Diagrams - STDs)
- Behavioural (such as Behaviour Diagrams - eFFBDs)
- Physical (such as Architecture Interconnect Diagrams - AIDs and Physical Architecture Diagrams - PADs)
- Process (such as Process Flow Diagrams - PFDs)
- OO, using UML (such as Use Case Diagrams - UCDs, Sequence Diagrams - SQDs, Collaboration Diagrams - CODs, Class Diagrams - CDs, Statecharts - SCDs, Activity Diagrams - ACDs, Component Diagrams - CPDs, and Deployment Diagrams - DPDs)
- Data relationship (such as Entity Relationship Diagrams - ERDs, and Data Structure Diagrams - DSDs)
- Software code organisation (such as Structure Charts - STCs, Software Architecture Diagrams - SADs)

No single notation can express all of the abstractions needed by a typical project. For example, UML cannot express system operational and functional time sequences (only message sequences between class instances and activity sequences within and between classes) and it cannot represent the details of system architectures (for example, the electrical connections between system components), DFDs and IDEF0 do not represent time sequencing well (IDEF0 has an implicit sequence whilst DFDs are conceptually in parallel) and eFFBDs do not show states and modes at all well (and without 3SL's extensions, cannot represent the system environment).

The power of the Cradle environment comes from the ability to:

- Combine notations in model topologies that meet the project needs
- Define a hierarchy of models within a domain

- 
- Specify which notations are required in a particular model
  - Link models to requirements, PBS, SBS, acceptance, emergent properties and so on
  - Have implementation-independent analysis and implementation-dependent design
  - Overlay system models with process models that show how the system will be used

Cradle also allows performance analysis of architecture and system design models. Operating on characteristics of behaviour and can be used before functions are allocated, and before simulation is possible.

Performance assessment is complementary to simulation, or execution (such as timeline analysis). It does not rely on defined system behaviour, and can be used at the outset to define performance budgets for subsequent design levels (apportionment). Later, it can be used to aggregate from actual values.

Performance is expressed as user-defined characteristics, such as:

- Bandwidth
- Utilisation
- Size
- Cost
- Data rate
- Staleness
- Weight
- Power

These can be sub-divided (for example to study both best and worst case conditions) and are represented as user-defined formulae, created from a built-in library, user-defined table interpolations or look-ups, and user-defined routines. Any number of characteristics can be defined, and associated with each diagram symbol.

Operating scenarios, real data, constraints and environmental effects can be included and varied during analyses. Performance data and results are stored with the design in the database for subsequent analysis.

Results can be reported, exported as CSV for processing by data analysis tools or spreadsheets, or can be graphed.

By being integrated into the models, performance data is cross referenced to performance requirements and original stakeholder needs and acceptance criteria.

## Information Reporting

Cradle provides an extensive set of tools designed to report database information in various formats and methods. Your output could include:

- Standard text or graphic reports
- Traceability matrix
- Metric reports
- Complex user-defined documentation

Cradle's Document Publisher tool adds markup into a user-defined Word template. When a document is generated, the markup is used to query the database for information that is loaded into Word and formatted

---

according to the styles, contents lists and indexes of that Word template. Embedded binary data can be loaded into the document using the Microsoft Document Object Model.

Arbitrarily complex tables, automatic subsections, embedded diagrams, paragraph and section numbering and self-referencing within the document are all specified in the template markup.

Documents can include indexes, lists such as Table of Contents, traceability / compliancy matrices, cross references, diagrams, text, embedded figures / tables, and external files.

A document register is maintained to record the generation of formal project deliverable documents and to reprint old documents. The register shows the project's status as the issue states of these key documents.

Output can be produced to various files formats, paper reports or published to websites.

---

---

---

# Projects

Describes the basic concepts for Cradle projects and the information in them.

## Cradle Projects

Everything that you do in Cradle is in the context of a project.

You can create any number of projects. Each project can contain an arbitrarily large amount of data. Each project can have any number of people working in it at the same time. A project comprises:

- A 4-character project code that uniquely identifies the project
- A project title
- A Project Database (PDB) containing information
- A project schema that describes the types of information in the project
- A set of User Profiles used to authenticate access to the project's database

All projects inherit an initial schema when they are created from a default schema supplied by 3SL and described in the Default Project Schema reference in the Cradle documentation set. You can change this default schema at any time.

## Project Databases

Information in a Cradle PDB is held in items. Items are created by the people working in the project or are imported into the project from other sources (such as Word documents or external software). Each PDB can hold any number of items.

Cradle provides multi-user, concurrent, access to all PDBs simultaneously. Any number of people can work within each PDB at the same time. Cradle provides access controls that you can configure to control who has access to specific items as well as the kind of access.

Cradle does not need concepts such as check-in or check-out or merging. In Cradle, you do not check-out one or more items and work on them in a personal workspace and then check them back into the database. In Cradle, you do not need to merge the work of several individuals, identifying and resolving conflicts and inconsistencies between their work.

With Cradle everyone is working in a particular project works within the same database at the same time. This eliminates costly and time-consuming overheads from personal workspaces and also guarantees that no inconsistencies can arise. Cradle provides concurrency control on items, which ensure that each item has its own set of locks to guarantee only one read-write access to an item at any time, with any number of simultaneous read-only accesses.

Cradle provides all integrity and consistency controls that you would expect with any large scale multi-user database. All such mechanisms, including transaction rollback and commitment, are managed internally and transparently.

---

## Items

Items are the basic units of Cradle PDBs. Each piece of information, whether it is a requirement, a function, a component, an interface or a risk is an item of a corresponding type.

All projects have the predefined item types:

- Change Requests
- Change Tasks
- Data definitions
- Diagrams
- Events
- Input conditions
- Requirements
- Specifications

Further item types are defined in all projects' initial schema, such as:

- DATA BLOCK
- DEFINITION
- DOC SECTION
- FEATURE
- ISSUE
- PBS
- REFERENCE
- RESULT
- RISK
- SBS
- STAKEHOLDER
- SYSTEM REQ
- VERIFICATION

You can have up to  $10^{78}$  items of each of  $10^{78}$  types in each Project Database. Each item contains up to  $10^{78}$  attributes, each of which can store or reference up to 1 TByte of any kind of data. Most of the attributes are held in what Cradle calls frames. The type of data stored in a frame is determined by its frame type. You can define up to  $10^{78}$  frame types.

Each item comprises:

- A type
- An identity that distinguishes this item from other items of the same type, such as a requirement number
- An instance that reflects the evolution of the item, such as a version number or a draft ID
- A set of attributes that contain the data for the item

- A foreground and background colour that is used to colour items of this type in Hierarchy Diagrams (HIDs) and in tree views of information
- A set of comments (called annotations) that may have been applied to the item
- An edit history recording details of each and every edit of the item, each comprising:
  - The date and time of the edit
  - Who performed the edit
  - The reason for the edit
  - The set of attributes modified by the edit with the old and new values of each

Each item has its own life history, reflecting its creation, evolution and its possible destruction. Items are typically created, modified, and passed through a series of formal reviews into a formal project baseline created with Cradle's built-in Configuration Management System (CMS). An item may stay baselined thereafter but normally it evolves. This evolution creates a new draft instance of the item as a piece of work-in-progress data. This draft instance is modified, passed through a set of formal reviews, and ultimately registered into a new project baseline as a new version of the item.

Over time many versions of any or all items in the project may be accumulated through a series of modify, review and register into baseline stages. All of these versions are instances of the item. Under normal circumstances, only the latest version of the item is accessible by members of the project and the previous versions are available only to privileged project personnel for reference. This can be controlled by the project as part of its access control policy.

A project can choose to record only the succession of versions of each item in successive baselines. A project can also choose that it wishes to record each and every edit to the item between these baselined versions as well as the series of baselined versions themselves. Cradle can record such edit histories for any or all items of any or all types. This is configured within the project's schema.

## Attributes

Items have attributes that describe the item and store its data. There are three kinds of attribute:

- Frames - large attributes that can be used to search and select items
- Categories - small amounts of data as free text, single value picklists or multiple value picklists
- Predefined - used to manage items, and include attributes such as name, number, version, owner, security classification and last modification date

The predefined attributes are always present. You can inspect their values, query the database to retrieve items based on the values of these attributes, and, in some cases, directly set the values of the attributes. Other predefined attributes are central to Cradle's access control mechanisms and the Cradle CMS. You cannot delete predefined attributes.

Categories are used to store small amounts of data. A project can define any number of categories, and then assign each category to none, one or more types of item.

Category definitions often have a list of legal values for the category, a pick-list. Such categories only allow users to choose one of the predefined values. If a category does not have a pick-list, then it can be set to any value.

Frames are used to store large amounts of data. Each item can contain any number of frames. Each frame can store up to 1 TByte of information.

---

## Categories

Categories are typically small attributes that are used to mark characteristics of items.

There are two types of categories, multi-value and single-value. Multi-value categories allow you to set multiple values for the category.

A project can define any number of categories, each consists of:

- A name that is unique among all categories
- Whether the category is mandatory or optional when it is used
- An optional picklist of values
- Optionally, for each value:
  - A set of category recognition strings, each a regular expression, that are used to scan text and find the value from the category that best corresponds to the text
  - A colour

A project can assign up to any 16 of these categories to each type of item. A category does not have to be assigned to any item types, it can be assigned to a single item type, or it can be used in multiple item types. This is simply a mechanism to allow common attributes to be defined once (as categories) and then used in whichever item types need them.

Typical category codes are:

- **Responsibility**
- **Priority**
- **Status**
- **Type**
- **Maturity**
- **Applicability**
- **Testability**

When assigned to item types, categories can be associated with other attributes in an item. When the value of the category changes, the associated attribute's colour is changed to the colour specified for the category value. For example, if a category **Maturity** is assigned to an item type requirement, then changing the maturity of a requirement could change its text statement from black to green.

## Frames

Most of the data held within items is held in frames. Most frames contain text, but a frame can store any type of data, and can also reference data held elsewhere. Frames that do not store text are called binary frames.

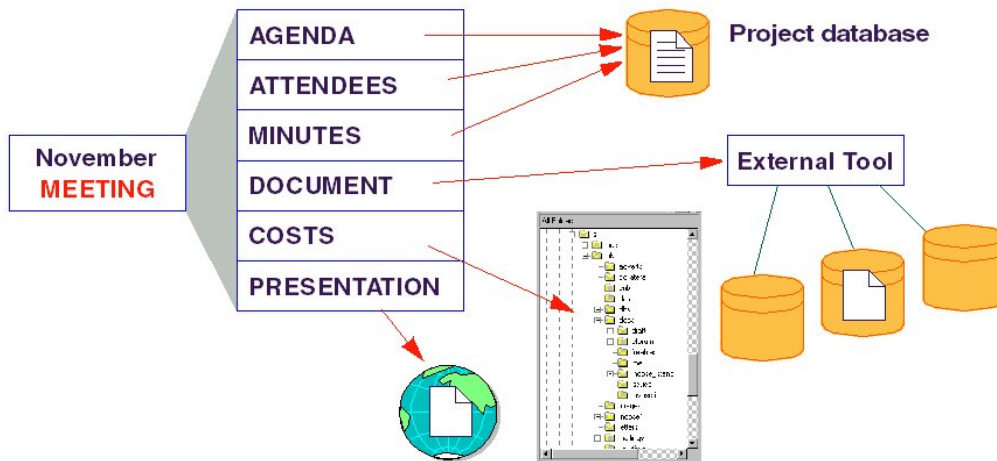
Each binary frame has an underlying frame type. Projects can define any number of frame types. Each frame type is a data type that describes a type of information to be held within all frames of this frame type.

This mechanism allows frames to contain any type of data, such as images, Word documents, Excel spreadsheets, PowerPoint presentations, audio files, movie files, or CAD drawings.

Frames either contain data, or reference it as:

- The pathname of a file held somewhere on your computer system
- The identity of data in another environment, such as a document or drawing control system
- The URL of a page on your project or company intranet, or the Internet

For example:



Here an item called **November** of type **MEETING** has six frames **AGENDA**, **ATTENDEES**, **MINUTES**, **DOCUMENT**, **COSTS** and **PRESENTATION**.

The **AGENDA**, **ATTENDEES** and **MINUTES** frames contain text and are stored in the Cradle Project Database. The other frames contain binary data, such as a Word document, an Excel spreadsheet, and a PowerPoint® presentation, respectively:

- The **DOCUMENT** frame is stored in another environment, such as the Documentum® document control system, and the frame stores the identity of the document in this environment
- The **COSTS** spreadsheet is stored as a file on a file server, and the frame stores its pathname
- The **PRESENTATION** frame's presentation is held in the Internet, and the frame stores a URL

## Cross References

Items can be inter-related by creating cross references between them. A cross reference between a pair of items means they are dependent on each other. You can link items simultaneously in any number of hierarchies, graphs or any other type of 1:1, 1:N or N:M relationships. Cross references also contain attributes, so you can store data inside each cross reference, such as:

- The reason for the link
- The justification for the link
- The current approval status of the link
- The coverage of the link

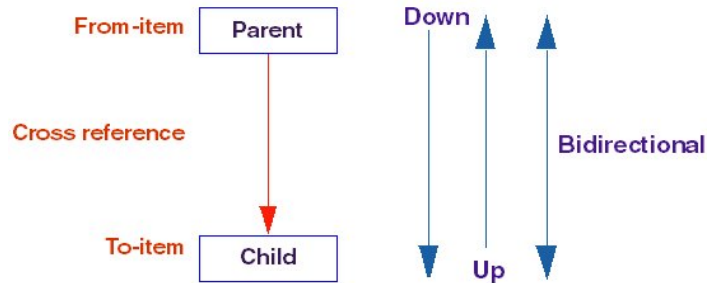
---

Each cross reference links a pair of items, the **from** item and the **to** item. Cross references can be followed from either item to find the item at the other end. The **from** item is usually called the parent and the **to** item is the child.

Cross references are controlled by link rules that allow you to control:

- What items can be linked
- Who can create the link

We define down to mean following cross references from the **from** item to the **to** item and up to mean from the **to** item to the **from** item.



Which results in the following search directions:

- Downwards** - follow the cross reference from the **from** item to the **to** item
- Upwards** - follow the cross reference from the **to** item to the **from** item
- Bidirectional** - both downwards and upwards

Cross references contain attributes:

- The **from** item, which item the cross reference comes from (such as a requirement)
- Predefined attributes, used internally by Cradle (such as the creator, creation date, last modifier and last modification date)
- The **to** item, which item the cross reference goes to (such as a test case or a function)
- Up to 16 user-definable attributes including:
  - A **Reason**, in which you can record why these items have been linked together
  - A **Rationale**, in which you can record the basis for the cross reference to exist, or what would cause the cross reference to become invalid
  - A **Reference**, in which you can record a reference to an external document's paragraph or table that prescribes the existence of the cross reference
  - A **Note**, in which you can record miscellaneous text, such as the review status of the cross reference as one of **New**, **Agreed** or **Disputed**
- A cross reference type, called a link type

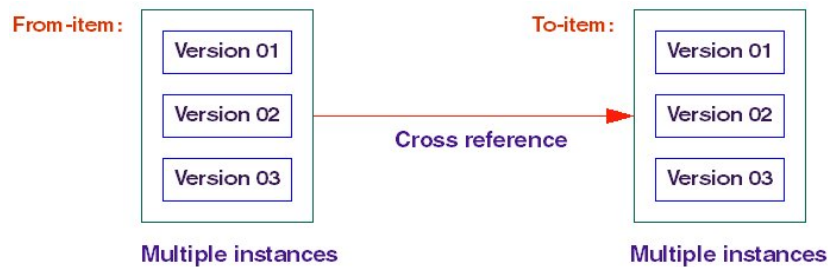
The user-definable attributes can be used to store information about the link itself. When searching the database you can, for example, choose to only follow cross references whose **Rationale** attribute has the value **Agreed**. You can define project-specific pick-lists for each of these attributes. If you do not define a pick-list for a cross reference attribute, then users can store any data in the attribute.

Cross references can be used:

---

- For impact analyses, finding the consequential changes needed for other items given that a particular item has changed
- For traceability, ensuring that all items in one set are linked to items in another set and vice versa (such as between user requirements and acceptance criteria, or between system requirements and test specifications)
- For change propagation
- Within tables and documents
- As navigation aids when browsing the database with any Cradle tool

Cross references are not specific to an individual version or draft. A cross reference effectively links all instances (versions and drafts) of the **from** item to all instances of the **to** item:

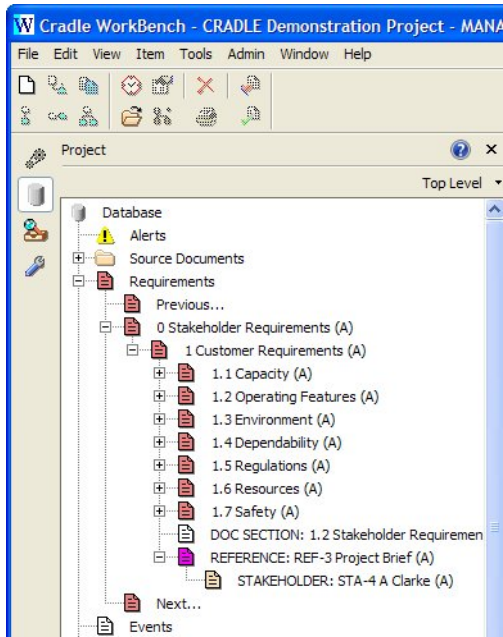


Cradle can present and manipulate cross references in a variety of ways, including:

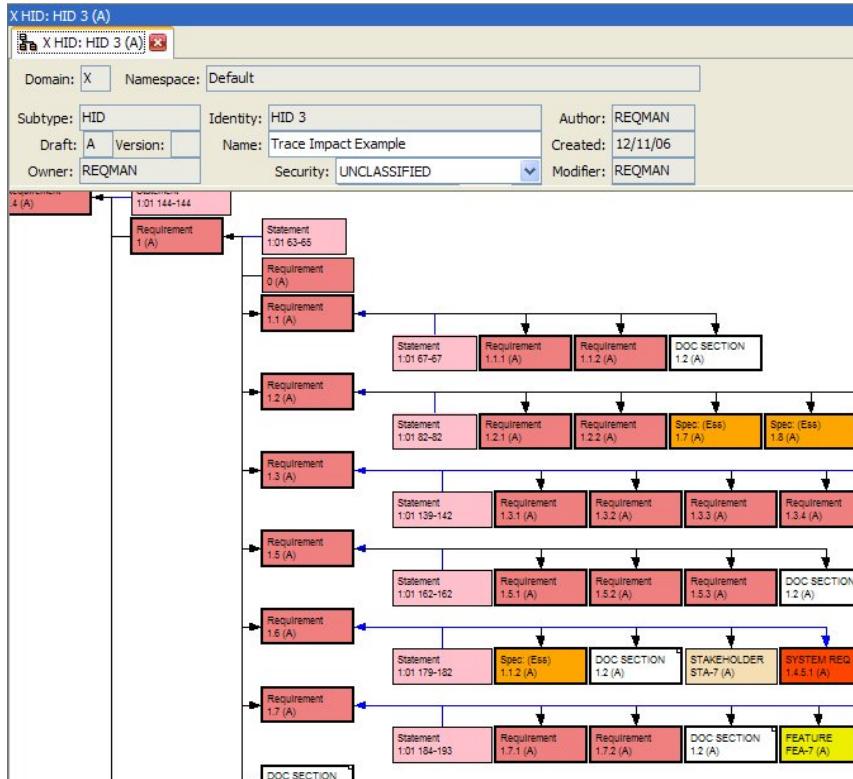
- Lists
- Tables, in which cross referenced items are shown in nested rows, creating tables that can be displayed, printed, and sent to Word, Excel and web pages:

QUERY: Requirements - All										
Requirements - All										
Identity	Name	Requirement Details	Status	[LINKED] System Requirements						
				Identity	Name	Status	[LINKED] System Breakdown Structure			
Previous...										
Identity	Name	Requirement Details	Status	Identity	Name	Status	Identity	Name		
1	0	Stakeholder Requirements	ACCEPTED							
2	1	Customer Requirements	AGREED	The outline requirements for the aircraft are:						
3	1.1	Capacity	AGREED							
4	1.1.1	Flight Crew	Low running costs is key to ensuring the service is competitive with alternative forms of transport. The costs of operating the service shall be minimised by using one pilot and one other flight crewmember, ensuring compliance with CAA guidelines and regulations.	CANDIDATE	1.4.3	Corrective Maintenance (Air)	REVIEWED	4.4.1	Fault Monitoring System	
								7.1	Pilot	
								7.2	Co-pilot	
					1.5.2.1	Accommodate Air Crew	REVIEWED	1.1.1.2	Window	
								1.1.1.4	Flight Deck	
								3.3.1.2.2	Cockpit Lights	
								7.1	Pilot	

- Explorer item hierarchies that can be navigated, viewed, and used in drag-and-drop operations:



- User-customisable Hierarchy Diagrams (HIDs) of items and the links between them that can be used to browse items and links, and present sets of attributes in vertical or horizontal hierarchies:



- Matrices, that show items as rows and columns in a tabular presentation, and where the cells in the table either show the existence of a cross reference and its direction (shown by the direction of the arrow in the cell):

MATRIX: Requirement Coverage

Requirement Coverage	1.1.1 : Flight	1.1.2.1 : Passenger	1.1.2.2 : Luggage :	1.2.1.1 : Cruise	1.2.1.2 : Runway	1.2.2 : Flight	1.3.1 : Wind : : A	1.3.2 : Temperatu :	1.3.3
1.1.1.1 : Change Direction : : A									
1.1.1.2 : Control Ground Speed : :					↖				
1.1.1.3 : Stop Aircraft : : A					↖				
1.1.2.1.1 : Monitor Course							↖		
1.1.2.1.2 : Adjust Heading : : A									
1.1.2.2.1 : Monitor Speed : :				↖					
1.1.2.2.2 : Adjust Speed : : A				↖			↖		
1.1.2.2.3 : Provide warnings									
1.1.2.3 : Control height : : A									
1.1.2.4 : Control attitude : : A									
1.1.2.5.1 : Control Weight :		↖	↖		↖				
1.1.2.5.2 : Control centre of		↖	↖						
1.2.1 : Monitor position : : A									
1.2.2 : Monitor height : : A									
1.2.3 : Monitor track : : A									

or which show the attributes inside the cross references between the items:

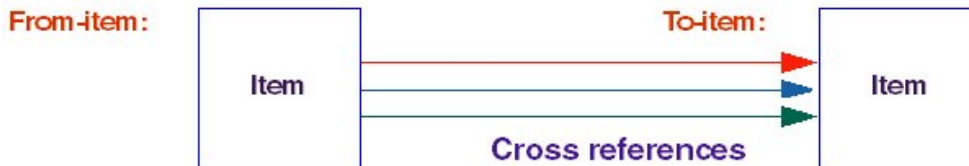
MATRIX: VCRM									
VCRM									
	1.2.1 : Bandit Standard Airframe : : A			1.2.2.1 : Integrated Air Power System : : A			1.2.2.2 : Propellor : : A		
	ID	V&V Method	V&V	ID	V&V Method	V&V	ID	V&V Method	V&V
1.1.1.1 : Change Direction : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready			
1.1.1.2 : Control Ground Speed : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready	VER-17	Test	Pass
	VER-13	Demonstration	Ready	VER-13	Demonstration	Ready			
	VER-17	Test	Pass	VER-17	Test	Pass			
1.1.1.3 : Stop Aircraft : : A	VER-1	Analysis	Pass	VER-1	Analysis	Pass	VER-17	Test	Pass
	VER-6	Inspection	Ready	VER-6	Inspection	Ready			
	VER-13	Demonstration	Ready	VER-13	Demonstration	Ready			
	VER-17	Test	Pass	VER-17	Test	Pass			
1.1.2.1.1 : Monitor Course Made Good : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready			
1.1.2.1.2 : Adjust Heading : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready			

or which show attributes of the one or more items linked by cross references to the items shown in the rows and columns:

MATRX: VCRM											
Requirements - All			Requirement Coverage			Product Feature Compa...			System Reqs - Acceptance		
	1.2.1 : Bandit Standard Airframe : : A			1.2.2.1 : Integrated Air Power System : : A			1.2.2.2 : Propellor : : A				
	ID	V&V Method	V&V	ID	V&V Method	V&V	ID	V&V Method	V&V		
1.1.1.1 : Change Direction : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready					
1.1.1.2 : Control Ground Speed : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready	VER-17	Test	Pass		
	VER-13	Demonstration	Ready	VER-13	Demonstration	Ready					
	VER-17	Test	Pass	VER-17	Test	Pass					
1.1.1.3 : Stop Aircraft : : A	VER-1	Analysis	Pass	VER-1	Analysis	Pass	VER-17	Test	Pass		
	VER-6	Inspection	Ready	VER-6	Inspection	Ready					
	VER-13	Demonstration	Ready	VER-13	Demonstration	Ready					
	VER-17	Test	Pass	VER-17	Test	Pass					
1.1.2.1.1 : Monitor Course Made Good : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready					
1.1.2.1.2 : Adjust Heading : : A	VER-6	Inspection	Ready	VER-6	Inspection	Ready					

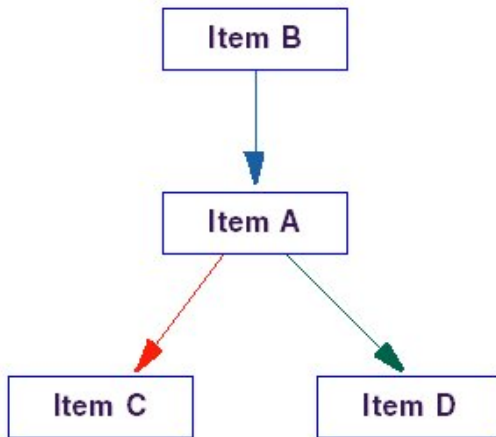
## Link Types

Different types of cross reference can be defined. These are called link types. You can have multiple cross references between the same pair of items, provided that their link types are different:



By default, all cross references have an empty link type. You can define any number of link types and then search the database for cross referenced items using only links of a particular type.

For example, if we search for items cross referenced bidirectionally to **Item A**, we find **Item B**, **Item C** and **Item D**. However, if we search using only **red** links, then we only find **Item C**:



We can define collections of link types, called link groups. If we define a link group **not-green** to be link types **red** and **blue**, and we search bidirectionally from **Item A**, then we find **Item B**, and **Item C**, but not **Item D** as it has a **green** link, which is not part of the link group.

Cross reference link types can be used in a number of different ways, to record:

- different types of dependency - With values such as: **Is required by, Is allocated to, Is an element of ...**
- significance of links - With values such as: **Critical, Very important, Important ...**
- status of links - With values such as: **Agreed, Disputed, New ...**
- variants and sub-variant dependencies - Using link groups for each product range, such as: Operating systems, Office applications, E-commerce and using link types for each product within the range, such as: NT4 and Word, Excel, PowerPoint and IIS, FrontPage

Cradle maintains multiple sets of cross references, one for each project baseline. This ensures that if a project ever reverts to a previous baseline, Cradle can restore the set of cross references that were in effect at the time of the (possibly very old) baseline.

## Link Groups

Projects can also define link groups, each containing a set of link types. You can then search the database with a link group, meaning that the only cross references to be followed are those whose link types are in the specified link group. This provides a very powerful and flexible mechanism.

These link types can be used to:

- Denote different types of dependency:
  - **Is required by**
  - **Is allocated to**
  - **Is an element of**
- Record significance of links:

- **Critical**
- **Very important**
- **Important**
- Record status of links:
  - **Agreed**
  - **Disputed**
  - **Pending**

Link types and link groups can be used to represent variants and sub-variant dependencies:

- Use link groups for each product range, such as:
  - **Operating systems**
  - **Office applications**
  - **E-commerce**
- Use link types for each model in the range, such as:
  - NT4 (Operating systems)
  - Word Excel PowerPoint (Office applications)
  - IIS FrontPage (E-commerce)

## Pseudo Cross References

Most dependencies between items are described by cross references in the database. Some items are related to other items not by separate cross references, but by a part of the item's own definition.

These internal relationships are called pseudo cross references. There are several types:

- Events to events
- Data definitions to data definitions
- Diagrams to the definitions of the symbols that appear in them
- Symbol definitions (such as specifications and data definitions) to the diagrams they appear in

For example, including the name of a data definition in a diagram creates a pseudo cross reference between the data definition and the diagram. Including the name of a data definition in a specification item creates a pseudo cross reference between the data definition and the specification. Including a specification in a diagram (such as a function in an eFFBD or a class in a UML class diagram) automatically creates a pseudo cross reference between the specification and the diagram.

Pseudo cross references are automatically created and destroyed by Cradle as users modify the contents of items in the database. They do not have a link type and do not contain attributes.

The differences between cross references and pseudo cross references are:

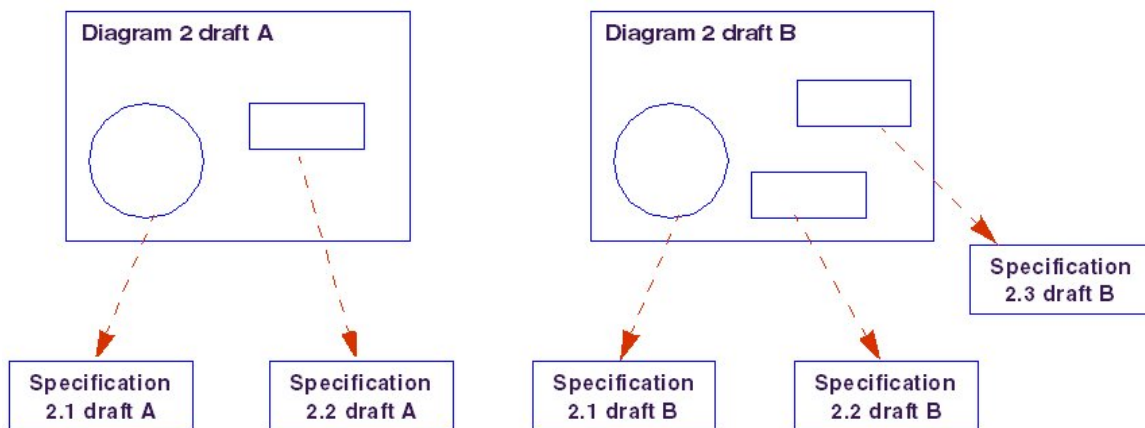
Characteristic	Cross Reference	Pseudo Cross Reference
Storage	Record in the database	No database record
Creation	Explicitly created separately from creating items (except for copying item hierarchies)	Automatically exist as soon as the items' content exists

Characteristic	Cross Reference	Pseudo Cross Reference
Contents	Set of attributes, some user-defined	No data content
Update	Contents can be modified separately from the items being linked	Exist, or not, depending on the contents of items, so changing the items' contents determines their existence
Deletion	Can be explicitly deleted (implicitly deleted if deleting an item hierarchy)	Automatically deleted if the linked items' contents change appropriately
Instances linked	All instances (versions and drafts) of the <b>from</b> item are linked to all instances (versions and drafts) of the <b>to</b> item	Links a specific instance (version and draft) of the <b>from</b> item to a specific instance of the <b>to</b> item

Cross references are records in the database that have attributes, some of which you can control. No matter what the content of an item is, without a cross reference being created in the database, items are not linked in any way.

For pseudo cross references, the relationship exists as a result of the content of one of the items. If this content is changed, the pseudo cross reference may disappear. Pseudo cross references do not have any component attributes. As they are based on the contents of items, pseudo cross references link specific instances of items together.

For example, a draft B diagram can link to more or different items than a draft A diagram. The pseudo cross reference to specification 2.3 exists only because the draft B diagram has within it a symbol representing this specification:



## Link Rules

Projects need to exercise the same care over the creation, modification and deletion of links as they do for items. Cradle allows projects to define an arbitrarily large set of link rules, each of which:

1. Either allows or prevents a specific operation
2. Applies to a specific user, or a group of users
3. Applies to a specific type of **from** item and to a specific type of **to** item

4. Optionally applies to only specific items of each of these types
5. Applies to creation and/or modification and/or deletion of cross references

This mechanism allows a project to have the same absolute confidence for the contents and integrity of the cross references within the database, as it has for the items in the database.

The definition and use of link rules allows projects to constrain the manipulation of cross references in accordance with whatever development process is being used by the project.

## Ownerships

Every item in a project has an owner; one of:

- The name of a user in the project
- The name of a team of users in the project
- **PROJECT**, meaning the project itself, when the information will be part of a baseline

Ownerships are used to define access rules; who has what kind of access to items in the database.

Projects need to identify their users, so that ownerships can be set and access rights determined. Each project has a user register that defines the members of the project as a set of User Profiles, each containing:

- A username
- A password
- Additional information including team name, full name, e-mail address, privileges and so on ...

To use a Cradle tool, you must identify which project you want to work in, and the user account that you want to use in that project. Hence, all of the Cradle tools require you to log-in to a project, by specifying:

- Username
- Password
- Project code

## User Profiles

Users have to log-in to a Cradle project to be able to work in its database. Each project has a user register containing the set of User Profiles (login accounts) that have been defined for that project. All projects have a User Profile called **MANAGER** (default password **MANAGER**) which is the most powerful User Profile for that project. The only part of the **MANAGER** profile that can be changed is its password.

Each User Profile defines the user's username and password, optionally the team to which the user belongs in the optional project organisational structure, and a set of privileges and access rights.

Projects can define rules for User Profiles, including password formats and password aging.

Cradle also supports LDAP protocol for login. Refer to the System Administration Guide for further information.

---

---