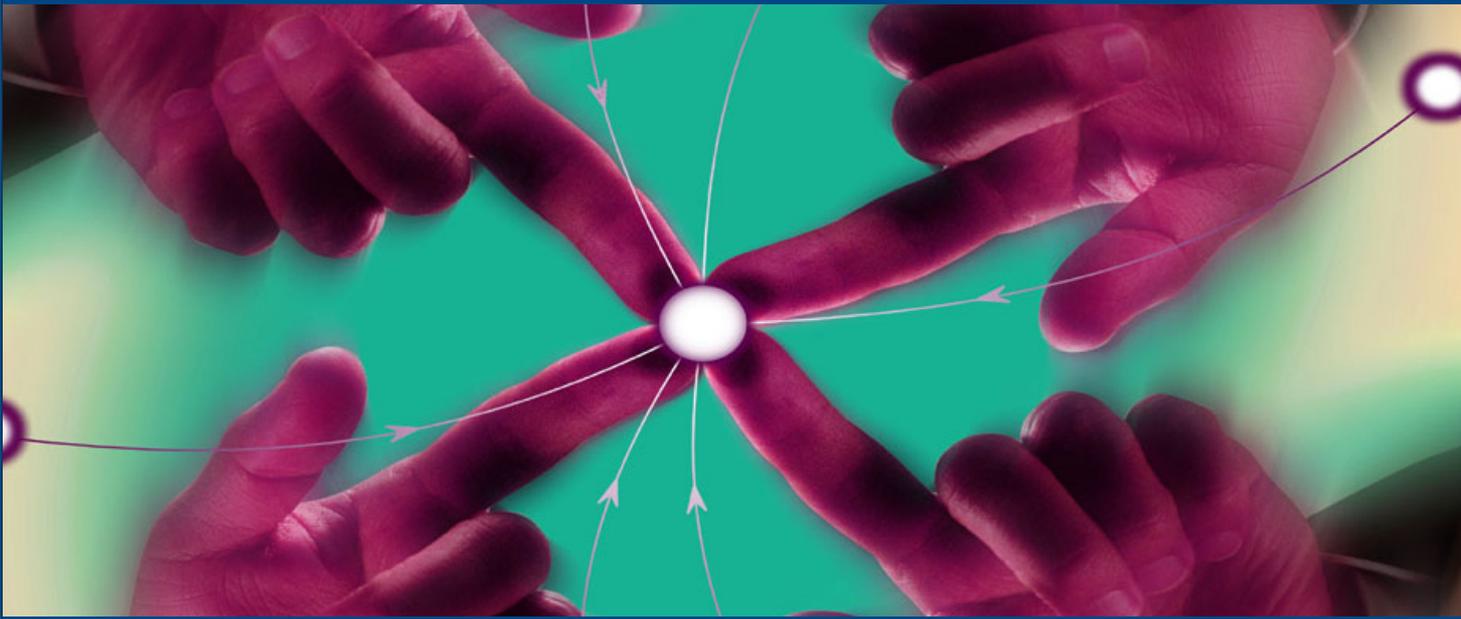


# AllChange v7.0

from Intasoft







# Fast facts

---

The functionality provided by AllChange includes change, configuration, process, release, and asset management within a single cohesive product. As a Windows-based product it was historically aimed at the management of development environments but more recently added features mean that the product now addresses a broad range of configuration management requirements, including document management and web development, as well as hardware configuration management.

## Key findings

In the opinion of Bloor Research the following represent the key facts of which prospective users should be aware:

- AllChange is extremely flexible and is designed to support whatever processes and procedures the user wants to implement.
- AllChange is a comprehensive solution delivered as a single integrated package rather than in a series of 'bolt on' modules. Flexible licensing schemes include reserved-named or concurrent licensing with sub-options for full, change request only, or read only (free) access.
- The product includes a comprehensive web browser interface that provides configuration management functionality equivalent to that provided by the Windows-based interface.
- The product provides extensive workflow capabilities, ensuring that no company-defined processes can be bypassed. We are particularly pleased to see the introduction of graphical lifecycle definition and generation capabilities in the latest release (7.0).
- IntaSoft is a small company. While we appreciate that this may put some potential users off, the fact that the company has a significant user base (for example, Lloyds TSB and Raytheon to name but two) and has been in business for nearly 20 years should reassure most prospective customers.
- A particularly strong feature of the product is its integration with Microsoft Office and its compliance with Microsoft's Common Source Code Control Interface. In the case of Office the integration with both Word and Excel is especially close. A number of other important integrations are also provided.

## The bottom line

AllChange is at least as feature rich as its major competitors and in some areas more so. In addition it is configurable to meet the needs of any organisation that wants a comprehensive change and configuration management tool. Add to that the fact that AllChange is significantly less expensive than other products in the



market and you have a compelling argument in favour of considering AllChange for the management of any suitable environment.



# Vendor information

---

## Background information

AllChange originally came about when two Exeter University students wanted to build a configuration management facility to support their own development efforts and Intasoft was founded, in 1986, to market and develop the resulting product.

The company has a substantial customer list which is peppered with household names (both end users like Westland Helicopters and systems integrators such as CSC and Fujitsu) and even before the company opened its first non-UK office (in Maryland) it had customers in the United States, as well as in Europe, Israel and Australia.

Intasoft works particularly closely with its customer base. It is a small company and depends heavily on a close liaison with its clients in order to ensure that its product development reflects the ongoing requirements of its users. Indeed, it is notable that although the company has historically focused on direct sales (it has only set up overseas distributors within the last year—see below) it has no sales force as such, with all of its key personnel being intimately involved with the development of the product. While this clearly has its drawbacks from a marketing perspective, it does mean that those involved with sales are of a higher calibre than is often the case. It is also noteworthy that the company is particularly proud of its excellent reputation as a training and consultancy provider.

Intasoft web address: [www.intasoft.net](http://www.intasoft.net)

## Product availability

The latest release (April 2004) of AllChange is version 7.0. There are two licensing schemes (Named and Concurrent) each containing 3 licensing options: a Full license, a Change Request only license (which provides full access to change management functions but is otherwise read only), and a Read-Only license for which there is no charge. Typical installations consist of a mixture of these different options so that you only pay for what you need. It should be noted that within the concurrent licensing scheme, where licenses are normally shared between users, you can opt to reserve licenses for administrators only to ensure that the administrators can always access the system when they need to.

AllChange runs only under the various Windows operating systems but uses FTP (file transfer protocol) to manage objects that exist in other environments. The company has partnered with a number of third party vendors within the development arena, and the product has been specifically integrated with Telelogic's DOORS, Artisan's Real Time Studio, Araxis Merge (in the latest release), I-logix's Rhapsody, Macromedia Dreamweaver, Mercury Test Director, and the Unipress Help Desk and Fault Reporting system. In the latest release, the integration with



DOORS has been enhanced so that AllChange change requests can now be created and updated directly from the DOORS environment.

In addition, AllChange complies with Microsoft's Common Source Code Control Interface which provides integration with Microsoft products such as Visual Basic, Visual C++ and Visual Studio, as well as other products that support this standard, such as Sybase's PowerBuilder and IBM's WebSphere Studio.

Where integration is required to a product that is not included in one of the categories above, then this can be accomplished on a site-specific basis. For example, an integration was built with Computer Associates Advantage OpenRoad software development environment specifically at the request of a particular customer.

Apart from AllChange, Intasoft also markets IntaChange, which is a web-based change management application that has only recently been released (March 2004). A separate review of this product is available from Bloor Research.

## Financial results

Intasoft is privately financed and financial figures are not published. Unusually in this day and age the company is neither VC-backed nor has it floated on any stock market. Moreover, the company has no plans to go in either of these directions. However, this does have the downside that the company is very small, with only 9 employees, which may be a concern for some potential users. However we do not think that this should be a problem. First of all, the company has been in business for the best part of two decades, and is continuing to thrive and, secondly, the way that the product has been designed has an impact on this. While we will discuss the details below, suffice it to say for now, that the configurability of the products is such that both IntaChange and AllChange will support existing working methods with only a minimal effect on the working environment. This means that, once the initial installation period is over, there should be a lower support requirement than might be the case with more prescriptive products.

In 2001 the company made a substantial move into overseas markets, and Intasoft now has an office in Maryland, USA, as well as a distributorship agreement with a partner in Germany (which has operations throughout continental Europe), as well as distributors in both Australia and China.



# Product description

---

## Introduction

Like most products, AllChange consists of core functionality that supports its major capabilities. In this case the core functionality consists of the version control system and what Intasoft refers to as the parts database (though you can call it whatever you like), while the product's major capabilities encompass configuration, change, process and release management. However, in addition to these two areas, a major feature of AllChange is its configurability and extensibility, which is exemplified during implementation.

We do not usually start a review of a product with a discussion of its implementation configurability. However, in our view this is AllChange's BIG advantage. The product may, indeed does, have some nice technical features. But so do other products. It also has price advantages when compared to its major competitors, but its big differentiator is its configurability. This is particularly important for a CM tool because, while you may want to manage and control the development environment, you may not wish to actually change that environment (though, of course, you can do that if you want to). The following sections on implementation detail generic configurability features, though there are also a variety of relevant facilities which are specific to particular technical aspects of the product (especially life-cycles), which are discussed in subsequent sections.

## Implementation

Unlike some of its competitors, AllChange is not modular. There is no separate configuration builder, version manager or developer's toolkit. Instead, all of the relevant functionality is integrated into the one product. So, one aspect of configurability is simply selecting those aspects of the product that you are going to use. More important however, are the levels of control to be implemented. This will vary from project to project. So configurability does not only apply at a site level but also to individual projects.

### User interface

AllChange includes a Configuration Editor that is used to set up and administer the implementation environment, of which there are a number of major elements. Some of these are specific (see later) while others are general-purpose. Those in the latter category include the user interface, the AllChange Command Evaluation Language (ACCEL), and the entry conditions and actions that may be applied to commands.

AllChange supports two different user interfaces: a Windows based client application that shares the Microsoft Outlook 'style'; and a web-browser interface. The only real difference between the two interfaces is that you can customise the thick client interface, for example by adding, removing or re-arranging columns,



which you cannot do within a browser. An example of the browser interface is illustrated in Figure 7 later in this review.

In addition the product is integrated with Windows Explorer, which allows right-click access to AllChange functions.

As far as the standard interface is concerned, this comes, as do the product's other components, in a standard configuration. That is, it is delivered with a pre-built out-of-the-box solution that enables you to get up and running as quickly as possible. However, in the latest release, the product also ships with a pre-built ITIL (IT Infrastructure Library) configuration as well as Intasoft's own. In other words, you now have a choice of which out-of-the-box solution you want to start with.

At a more detailed level, both menus and toolbars can be tailored on either a site or an individual basis, while additional functions and dialogues may be added through use of ACCEL. Sets of commands and parameters can be defined to produce a standard look and feel and, in the latest release, you can also define your own nomenclature. Thus, for example, if you want to use "RFC" (request for change) or "Trouble Ticket" rather than "CR" (change request), which is the Intasoft standard, then you can change this as appropriate.

While on the subject of standard configurations it is also worth noting that AllChange is delivered with a tutorial application that sets up a tutorial database and auto-configures AllChange to its standard configuration. It would be useful if Intasoft was to implement a similar capability for its other out-of-the-box (ITIL) solution. There is also a configuration utility for which the company plans to provide downloadable tutorial lessons.

The other major interface supported by AllChange is with Microsoft Office, with which the product has been closely integrated. In the case of Word, for example, new files may be submitted to AllChange, and existing files can be checked in or out, directly from Word. In addition, you can embed any AllChange data field into a Word document, which will be dynamically updated as the data field changes. These facilities are driven by Word Macros which are supplied as both menu options and toolbar buttons (to be implemented as required). In addition, there is a Word wizard which allows AllChange reports to be generated as new Word documents.

Integration with Microsoft Excel and Microsoft Project are discussed further later, although it is worth mentioning here that Excel is integrated to the same extent as Microsoft Word, including the ability to check in and out from these applications. As we mentioned above, AllChange is also compliant with Microsoft's Common Source Code Control Interface standard and this provides the ability to check files in and out of AllChange directly from supported development environments (see Product availability section).

### Commands

AllChange commands can have both entry conditions and actions associated with them on a per site basis. The former may be used for access control purposes though the utilisation of user roles, which is provided as standard, means that



this will seldom be required. However, entry conditions can also be used to define other sorts of pre-conditions. For example, you might require a change request to be authorised before particular types of work can be carried out.

Command-related actions are also defined on a site-by-site basis and provide a substantial degree of configurability. Examples that can be defined include one action triggering another, or an action causing an e-mail to be sent.

### **Interoperability**

There are two levels at which the configurability of AllChange impacts on interoperability. The first is through use of ACCEL and the second is by utilising the command-related actions which were discussed above.

ACCEL is the built-in programming language. It provides access to fields within the database as well as links into AllChange functions and commands. It also supports the usual sorts of conditional functions, variables and so on. Thus it can be used to build complex reports that go beyond the scope of the built-in report generator (see later). More particularly, it is extensible and provides access to both the GUI and operating system, including support for various means of exchanging information with other programs, such as DDE and OLE as well as FTP. There is also a published API. Thus, in conjunction with command-related actions, ACCEL can be used to integrate AllChange with other environments.

The MS Project interface is only one-way (because that's what customers want) with features such as the ability to export CR information to Project, so that the progress of these can be monitored therein.

Finally, there are also interface options for reporting and data presentation. In the latter category there is the integration with MS Excel (there are basic 2-D and 3-D graphical capabilities within AllChange) while, for reporting, there is an ODBC link available into the database so that third-party report generators can be used, if required.

## **Core functionality**

The heart of AllChange is the parts database (the other major core capability is the implementation of Change Requests, but this is specific to change management). From a logical perspective it makes sense to break down the consideration of the parts database—which, in effect, provides an asset management capability—into its constituent elements, which are discussed in the sections that follow.

### **The Parts database**

Products typically consist of a variety of items which may consist of on-line files such as source modules, documentation, design specifications and so forth, or may be external objects such as hardware. These elements, of whatever type, are referred to by Intasoft as “parts”. Parts are typically thought of as existing in a hierarchy of components, subsystems and so on. However, a more appropriate comparison would be to a bill-of-materials, since various relationships exist which will not fit neatly into a tree diagram. In particular, AllChange supports two ad-



ditional forms of relationship that might be described as Indirect. The first of these is when parts in one section of a configuration affect parts elsewhere. This is illustrated in Figure 1.

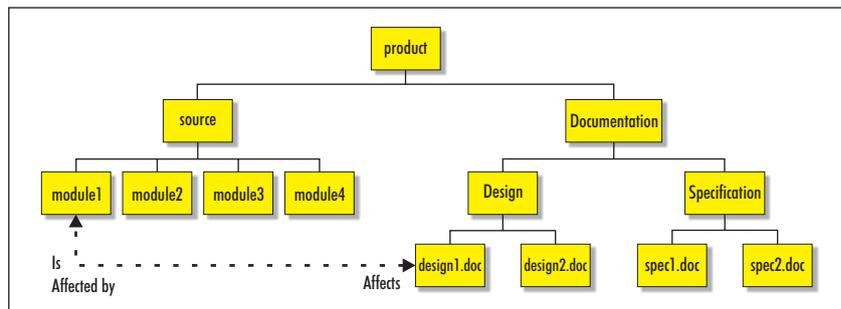


Figure 1: Parts affecting Parts

It should be clear that it is important to be able to cross-relate parts in this way, since you need relationships like these to be reflected when you use AllChange's impact analysis capabilities, for example.

A second form of indirect relationship that is supported by AllChange is usage relationships. In this case, a set of parts is reused in multiple places within a configuration, and the software allows you to define parts in this way so that they can be reused across product structures. An example of such an arrangement is illustrated in Figure 2.

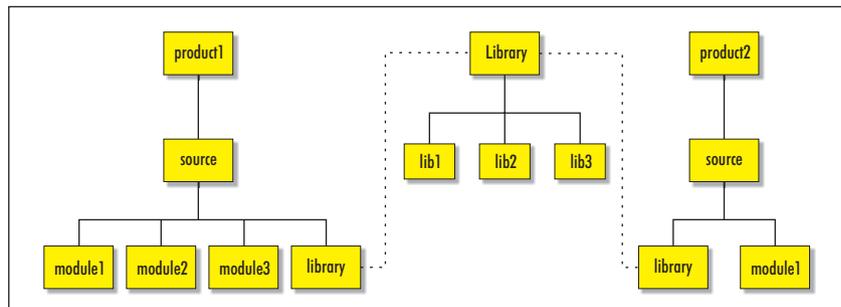


Figure 2: Usage relationships

The underlying database that supports these structures is based on FairCom's C-TreePlus. This is a well-known American database which uses ISAM techniques. Its major advantage from Intasoft's point of view is that it can license the database without having to pay any usage charge, which means that the company does not have to pass any additional fees on to customers. However, an additional licence is required from FairCom to use ODBC access.

### Version control

All parts may be versioned and AllChange supports the use of both deltas and full versions for this purpose. In the case of deltas the company uses merged deltas, the idea of which is to provide uniform access speeds, regardless of the version that is being retrieved. Typically, deltas are most useful when there is a large textual element involved but, as with other aspects of AllChange, this is under user control.

In other respects, the version control facilities are more or less what one would expect. There is a complete logging facility which details when a change was



made, who made it and why, as well as the actual storage of versions and their numbers. There is also the ability to be able to revert to any previous version. Version branching is supported (which is equivalent to supporting variants) and one nice feature that is provided is the ability to apply meaningful names to branches (as opposed to version 1.1.2.a.1, say, which is pretty much useless).

### **User facilities**

User facilities in AllChange include user roles, workspaces, registrations and alerts.

User roles are employed in a conventional way to implement access controls, although we would prefer it if users were only presented with options that they are entitled to use, rather than presenting all options and then preventing them from selecting those that they are not entitled to. The reason that the software has been implemented in this way is because the permissions are dynamic, though that does not preclude non-presentation of options even if it makes it more complex.

Inheritance is supported for role administration and individuals may have more than one role, while security may be implemented down to the level of individual commands and objects if required. In addition, the same rules may be used to enforce user-defined procedures. These may be combined, if this is deemed appropriate, with life-cycle status information (see the Process Management section).

AllChange is able to handle any type of configuration item regardless of whether it is file-based such as a document, or non file-based such as a hardware asset. When a configuration item needs to be updated it is checked out from the database (usually by the person who has been charged with carrying out the work) and, for file-based items, typically to their private area (workspace) on their file system. Workspaces can be shared where team collaboration is required. All check outs from the database are recorded, which means that anyone can find out what has been checked out for whatever reason, by whom and to where, which is very useful for impact analysis purposes.

In addition to private and/or shared workspaces, developers may also have access to shared pools, which consist of parts that have been published by their respective developers rather than parts that are still under development. This publishing may be controlled through a variety of processes and procedures, in the user-defined manner that characterises this whole product. One further possibility is to pass any completed or revised part to a second workspace for quality review, prior to publication. This process may apply prior to returning a part to the parts database (where it will be assigned a new version number automatically), as well as to inclusion in a pooled workspace. Also noteworthy is the ability to automatically update a workspace with the latest versions of the items that the workspace contains. This facility enables users to be informed that later versions of the items they have checked out to a workspace exist and gives them the option to update the workspace or not. This is useful for both software development and documentation where public folders need to be kept up to date.

In effect, the product is providing workflow capability here (and, indeed, comparable facilities are provided throughout the product). We are particularly pleased



that these workflow capabilities are now graphical and diagrammatic, which was not the case in the past.

It should be clear that this approach provides complete flexibility with respect to the development environment. You can, for example, insist that testing takes place at whatever stage(s) suit your environment. Similarly, you can allow parallel development or prevent it. In the case of the former, there is a visual merge tool (which has been enhanced in the latest release) which supports the reconciliation process between divergent versions. There is also the option to use the Araxis Merge product, with which AllChange has been integrated.

Another facility available to developers is the ability to designate particular versions of parts as a default requirement. This is what Intasoft refers to as registration. Use of this means that a developer can ensure that a particular set of parts is automatically loaded into any particular workspace whenever it is started up. This should ensure consistency, saves time and reduces errors. Note that different users may have different registrations. This can be quite important when it is necessary to isolate a particular user from changes that may be made by others in the same team.

The other facility that users have is to define alerts (though AllChange calls them monitors). These can be defined to send an e-mail (all the leading standards are supported) to the relevant user, whenever a particular event occurs. More or less anything can be monitored, including files, parts, baselines, changes and so forth, and monitored events may be either general, such as part updates or CR status changes, or they may be specifically designed to look for particular events.

## Process management

Process management in AllChange is all about life-cycles. While workspaces provide the underlying environment within which developers operate, life-cycles provide the underpinning for all the concrete elements of the environment such as parts, CRs, baselines (see below) and so on.

The definition of life-cycles is the other major way in which AllChange may be configured to the working environment. In effect, the definition of these life-cycles is the equivalent, in modelling terms, of defining a state transition diagram. That is, you describe a series of status definitions and then describe the conditions that allow one status to be converted to another. In addition, status changes can trigger other events. So, for example, a CR might have to be actioned and assigned to a particular user (life-cycles may be used in conjunction with user roles), before source code editing can take place. On the other hand, once editing is complete, the life-cycle definition can ensure that the part is checked back into the parts database.

In fact, the support for life-cycles is probably the area in which the most significant new features have been introduced in this release. In particular, it sees the availability of an Editor and a Lifecycle Generator, which are illustrated in Figures 3 and 4 respectively. In the case of the former, facilities include the ability to assign pre-defined conditions and actions to each status simply by clicking

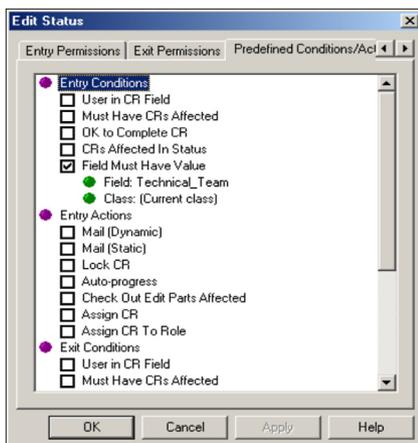


Figure 3: The Editor screen

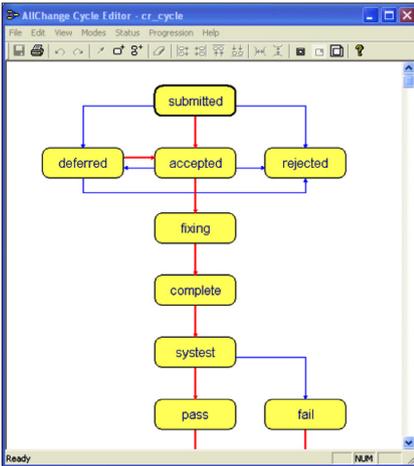


Figure 4: The Lifecycle Generator

on the relevant tick box. As an example, the “Predefined Conditions/Actions” tab allows you to enter rules and configure the workflow more easily than was previously the case, because you can build a rule, for instance, to say that you can only move from “CAB” (Change Advisory Board) to “Approved” if the field “Technical Team” is completed - simply by clicking on the relevant check box and selecting the appropriate field.

What the graphical Lifecycle Generator provides is the ability to create lifecycles by drawing statuses and linking them via progressions. Alternatively, you can define lifecycles conventionally and the software will generate the appropriate diagram automatically, which means that, if you are an existing AllChange user, you will be able to use the generator in conjunction with any existing lifecycles.

On an ongoing basis, users may also view the current condition of a lifecycle graphically. In this case, the diagram presented to the user will be essentially the same as that in Figure 4, except that each of the status boxes will be colour coded so that you can easily identify the current position in the lifecycle and you can progress from one status to another by clicking on the relevant box.

Apart from the use of life-cycles within individual elements of AllChange, it is also important to appreciate how process management can enable integration across the product, and this is one of its most important features. As an example, Figure 5 illustrates how change requests, configuration item processes and release management may all be linked together.

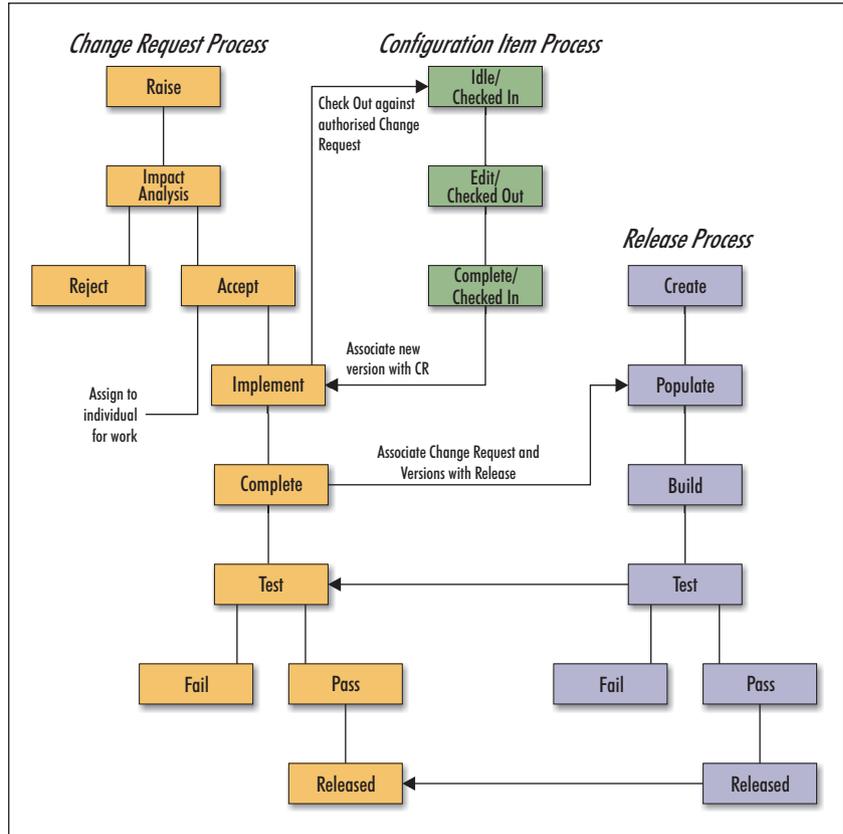


Figure 5: Linking the processes



This illustrates not only how AllChange can be tailored to the user's needs, but it also shows how life-cycles can be employed to automate many of the mundane but necessary processes that make up the development environment.

Life-cycles do not necessarily apply to say, CRs, en masse. Parts, CRs and so forth may be classified into sub-groupings with a common life-cycle. Thus, in a simple environment, all CRs might have a common life-cycle. In a more complex scheme, life-cycles might even be individually assigned.

## Product design & development

As we have noted, AllChange supports a number of different environments. However, there are facilities that are specific to software, particularly in the use of what Intasoft calls build threads (BTs), which are used to build a software configuration. This is essentially similar to the UNIX Make facility. A "desired BT" defines the versions of the parts that are required, together with any translation rules that are necessary. This ensures that an object is only re-built when its BT does not match the desired BT. This approach means that AllChange can distinguish between objects built with different compiler flags and can differentiate between objects built from different versions of the same source code. More specifically, multiple software configurations may be supported through the definition of appropriate build threads.

While a BT defines the construction of a product, a baseline defines its status at any particular point in time. To put it another way, a baseline provides a snapshot of a particular deliverable at a particular time. If that is a whole product, it will consist of the version numbers associated with the subsystems and components that make up that product. However, it is equally feasible to baseline an individual component so, more generally, a baseline may apply to any part. At the other end of the scale, large development projects may well be split into discrete sections, each of which will have its own baseline. These in turn can be baselined, so that one can have a baseline of baselines or, more accurately, a meta-baseline. In other words, a part baseline establishes a set of compatible versions while a meta-baseline defines a release, as illustrated in Figure 6.

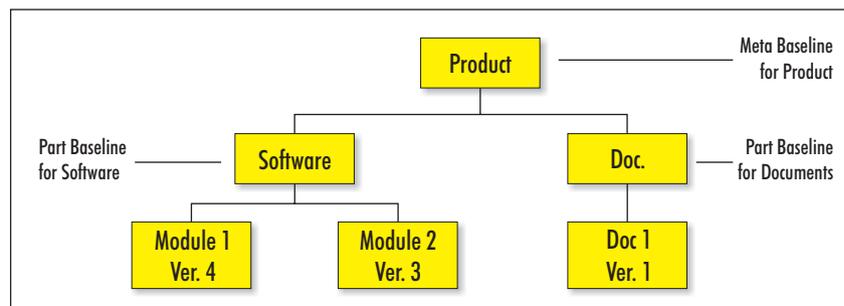


Figure 6: Baseline hierarchy

There are a variety of additional features. In particular, you can build a baseline in many different ways. For example, you can take a snapshot based on specified selection criteria, you can create and maintain an incremental baseline, and you can build release baselines on top of design baselines by applying similar criteria.



Baselines can also be used to provide both functional and physical auditability. Further, you can build a baseline based on the changes that you wish to implement—this is an important feature as it passes control of the release back to the business. Changes are written in business English and it is easier for business managers to think in those terms than try and get technical people to explain the impact of including or excluding a particular module.

Finally, in so far as products are concerned, these will need to be released, and AllChange includes a relevant manager. This creates copies of releasable objects which are copied from the appropriate workspaces (shared pools) into a release directory. Rules can also be applied to ensure that releases conform to corporate guidelines, while the release process can be automated using the baseline life-cycle in order to ensure that the right Configuration Items are delivered to the right places with a single click of a button. The integration of the business processes with the automation aspects of the product help to remove human error and guarantee that the right version of the right Configuration Item is delivered.

## Change management

Arguably, change management is the central feature of AllChange, despite the fact that we have left it, virtually, to last. Change management covers the whole area of fault reporting, bugs, error fixes and so on. However, its significance is that it is often the initiator for other AllChange activities. For example, you may prevent work from commencing until an appropriate CR has been raised. Like all of AllChange's facilities, the user can decide how he wants to employ CRs, but they are often used as the driver for the whole environment.

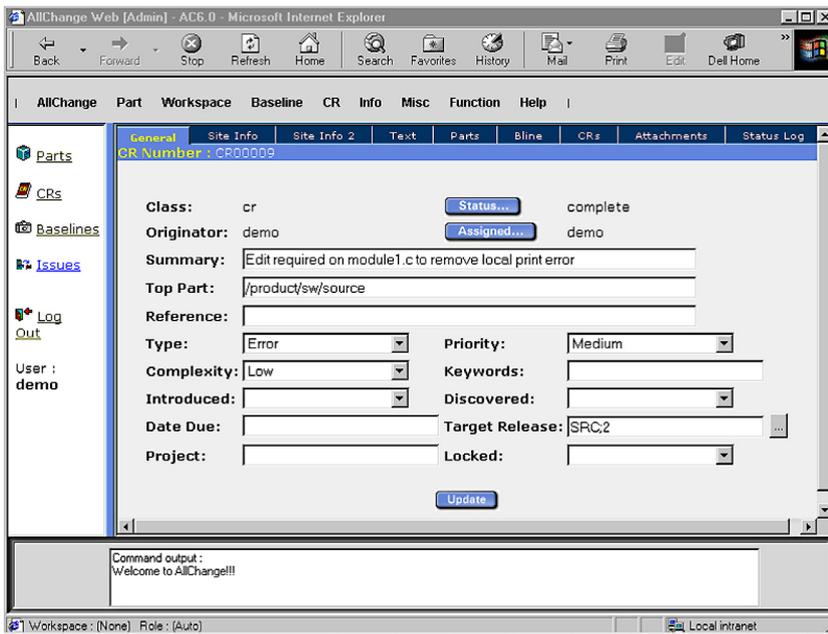


Figure 7: Browser-based screenshot

Like other elements within AllChange, CRs have an associated life-cycle, and the discussions in the section on Process Management apply in the same way here. In addition, each CR is assigned a unique identifier, according to whatever site standard the administrator defines. Different schemes for numbering may be assigned to different classes of CR, which have different life-cycles.

Further, CRs may be associated with parts, baselines, other CRs and, in this release, with meta-baselines. In the former case (which is illustrated in the browser-based Figure 7) they can be used to control changes to the relevant part while, in the latter, the conjunction of baseline and CR ensures that change is an integrated part

of the development process. This also provides traceability from the idea that generated the change through to the actual change itself.



As far as this screen-shot is concerned it is also worth noting the various attributes associated with the change request (that is, the type, priority, complexity, keywords and other elements in the lower half of the screen). These are what Intasoft calls arbitrary fields. While some of these are provided out-of-the-box by Intasoft, there are some 40 of these altogether and each of them may be user defined. Moreover, fields may be of various types, including drop down, combo and check boxes, date fields, hyperlinks and text. You can also attach such things as documents or screenshots to a change request. Actually, the use of arbitrary fields (for which there are now 40 wherever these are available—parts, versions and baselines) and attachments is implemented throughout the product, for example in release management and version control. This would allow you, for example, to allow hardware items to have unique serial or part numbers, associated costs and so forth.

As has already been mentioned, CRs may be used in conjunction with triggers and alerts in a number of other circumstances. CRs can also be cross-referenced across life-cycle classes. This enables a hierarchy of CRs to be developed, so that they may be cascaded. This is quite an important feature. For example, suppose that a job comes into a section: this generates one or more tasks for team leaders to perform and each task in turn generates CRs that actually cause the change to the physical Configuration Item. You can then build a rule that says that no task can be completed until all the CRs it has generated have been completed, while no job can be completed until all the underlying tasks are also complete. You can then use the MS Project interface to replicate this as a hierarchy of tasks and activities and produce a first cut Gantt chart.

Priorities may be assigned to CRs (see Figure 5) though there are no standard facilities for building an escalation procedure. You can, however, define events to ensure that CRs are dealt with in a timely manner, with alerts to inform relevant personnel if that does not occur, which may be delivered by third party scheduling products. In addition, there are standard reports to provide information about who has been issued with a particular CR, when it was issued, its timeframe for delivery and so on.

## Other facilities

The other major facility in AllChange is the report generator. This has already been mentioned and it can be used for both formal and ad hoc reporting. In fact, this was significantly enhanced in version 6.0 and this release only sees some minor additional features. It is reminiscent of (Business Objects) Crystal Reports and provides wizard-driven functionality with drag-and-drop for report creation, and support for features such as conditional options. In addition, it is worth noting that you can print the contents of the various browser windows as well as life-cycle diagrams. Since users can customise which columns (and, therefore, data) they wish to see in the browser, this mechanism can be used to print the required information as an alternative to generating a report.

Although not a part of the report generator per se, the visual differencing facilities of the product could also fall within this area. These are quite comprehensive



and provide both inter- and intra-file differencing. Note that you can compare any two versions, which means that the facilities provided are significantly more comprehensive than the tracking features in Microsoft Word, for example. It is also possible to compare baselines. This is valuable because comparing baselines enables you to see what has changed between two releases/deliverables and to drill down within the hierarchy of baselines, to get the differences between meta-baselines, sub-baselines and so on, eventually to the difference between the versions that went to make up the release. This is particularly useful when software development is outsourced and you wish to compare what is being delivered to you with what preceded it, or where a release is made and errors are reported as a result. Comparing the definition of the release just made with the one that was in place previously helps narrow down the area of investigation.

To provide these differencing capabilities Intasoft has integrated AllChange with Araxis Merge and, in fact, Intasoft acts as a reseller for Araxis. Alternatively, it is very easy to use third party differencing products (such as WinMerge) if you prefer.

## **Summary**

If AllChange was developed and marketed by one of the leading vendors there is no doubt that it would be a market leader in its sector. The fact is that it is actually produced by a small company based in the West Country. Nevertheless it has acquired major customers within government (both in the UK and Australia), within the finance community (for example, Lloyds TSB), and in the defence industry (Raytheon and others) amongst others. The product is also used worldwide by Learning Tree international in its IT training courses. All of this points to the fact that AllChange has something going for it. Indeed, that there is more going for it than you might expect from a relatively unknown company. If you are in the market for a change or configuration management system, or even asset management, then AllChange is certainly worth a look.

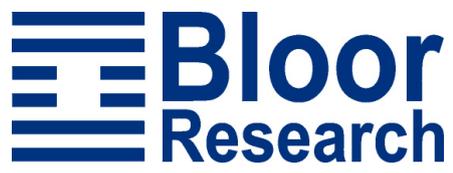
# Copyright & Disclaimer

---

This document is subject to copyright. No part of this publication may be reproduced by any method whatsoever without the prior consent of Bloor Research.

Due to the nature of this material, numerous hardware and software products have been mentioned by name. In the majority, if not all, of the cases, these product names are claimed as trademarks by the companies that manufacture the products. It is not Bloor Research's intent to claim these names or trademarks as our own.

Whilst every care has been taken in the preparation of this document to ensure that the information is correct, the publishers cannot accept responsibility for any errors or omissions.



Suite 6, Challenge House, Sherwood Drive,  
Bletchley, Milton Keynes, MK3 6DP, United Kingdom

Tel: +44 (0)1908 625100 – Fax: +44 (0)1908 625124  
Web: [www.bloor-research.com](http://www.bloor-research.com) – email: [info@bloor-research.com](mailto:info@bloor-research.com)